

**TECHNICKÁ UNIVERZITA V LIBERCI**

Fakulta mechatroniky, informatiky a mezioborových studií



**Diplomová práce**

Liberec 2010

**Bc. Radek Pšenička**

# **TECHNICKÁ UNIVERZITA V LIBERCI**

Fakulta mechatroniky, informatiky a mezioborových studií

Studijní program: N2612 – Elektrotechnika a informatika

Studijní obor: Informační technologie

## **Kooperace činností robotů ve funkci robot-snímač a robot-manipulátor**

**Cooperation of Robot Activities in Function  
Robot-sensor and Robot-manipulator of objects  
with sensor MetraLight for manufacture robots**

### **Diplomová práce**

Autor: Bc. Radek Pšenička  
Vedoucí práce: doc. Mgr. Ing. Václav Záda CSc.

#### Rozsah práce a příloh

Počet stran textu:	83
Počet příloh:	1
Počet obrázků:	57
Počet tabulek:	6

V Liberci 21.5. 2010

# TECHNICKÁ UNIVERZITA V LIBERCI

Fakulta mechatroniky, informatiky a mezioborových studií

Ústav mechatroniky a technické informatiky

Akademický rok: 2009/2010

## ZADÁNÍ DIPLOMOVÉ PRÁCE

Jméno a příjmení: **Bc. Radek Pšenička**

Studijní program: N2612 - Elektrotechnika a informatika

Obor: Informační technologie

Název tématu: **Kooperace činností robotů ve funkci robot-snímač  
a robot-manipulátor**

Vedoucí práce: **doc. Mgr. Ing. Václav Záda CSc.**

Zásady pro vypracování:

1. Seznamte se s programovacím jazykem Rapid a uživatelským prostředím ProgramMaker a dále se snímačem MetraLight a jeho užitím.
2. Navrhněte vhodnou koncovou hlavici robota, opatřenou jednak držákem snímače MetraLight a jednak držákem přídatného chapadla.
3. V programu Rapid realizujte úlohu vzájemné koordinace pohybu dvou robotů. Jeden robot bude sloužit jako mobilní nosič snímače MetraLight, druhý robot bude obstarávat požadovanou manipulační, či jinou činnost. Přídatné chapadlo nosiče prvního robota bude využito pro specifické činnosti.
4. Realizujte systém částečného rozpoznávání předmětů manipulace, z hlediska stanovení polohy, orientace a velikosti předmětu s určování tvaru.
5. Navržené a realizované zařízení testujte na souboru reálných úloh. Popište a případně statisticky vyhodnoťte přesnost a spolehlivost celého systému.

Rozsah grafických prací: dle potřeby dokumentace

Rozsah průvodní zprávy: cca 40 až 50 stran

Seznam odborné literatury:

[1] ROBOTSTUDIO 3.1 USER'S GUIDE. *Firemní dokumentace k programu RobotStudio*, Švédsko 2004.

[2] RAPID REFERENCE MANUAL. *Firemní dokumentace k programu Rapid*, Švédsko 2004.

[3] METRALIGHT REV.001 USER'S GUIDE *Firemní dokumentace ke snímači MetraLight*. <http://www.metrалight.com>

Vedoucí diplomové práce: doc. Mgr. Ing. Václav Záda, CSc.

Konzultant:

Zadání diplomové práce: 15.10.2009

Termín odevzdání diplomové práce: 21. 5. 2010

prof. Ing. Václav Kopecký, CSc.

děkan

V Liberci dne 21. května 2010

doc. Ing. Petr Tůma, CSc.

vedoucí ústavu

## MÍSTOPŘÍSEŽNÉ PROHLÁŠENÍ:

Byl jsem seznámen s tím, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé diplomové práce pro vnitřní potřebu TUL.

Užiji-li diplomovou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL. V tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Diplomovou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím diplomové práce.

V Liberci dne 21.5. 2010

.....

Radek Pšenička

## Poděkování

Na tomto místě bych rád poděkoval vedoucímu své diplomové práce, doc. Mgr. Ing. Václavu Zádovi, CSc. za jeho přístup, vstřícnost a cenné rady při tvorbě práce. Dále také své rodině a přítelkyni za nekonečnou podporu při studiu a především při tvorbě této práce.

Děkuji!

# Abstrakt

Cílem této diplomové práce je vytvořit funkční robotický systém pro rozpoznávání předmětů manipulace za pomoci laserového snímače. Během celé činnosti řízení systému budou všechna použitá robotická zařízení provádět koordinovaný pohyb. Tento systém může sloužit jako alternativa při typických průmyslových úlohách rozpoznávání předmětu v prostoru dle jeho tvaru a dále s určení jeho rozměrů, orientace a polohy v prostoru.

V úloze jsou použity dva průmyslové roboty od firmy ABB s typovým označením IRB 140 a IRB 1400. Pro detekci předmětů je využit laserový snímač MetraLight s typovým označením Rx series, který byl připevněn rameno robota IRB 1400. Dále jsou použity předměty pro rozpoznávání ve tvaru: válec, koule, kužel, krychle, pravidelný čtyřboký jehlan, hranol s podstavou čtverce, nepravidelný čtyřboký jehlan, hranol s podstavou obdélníka, válec položený na bok, trojboký hranol, trojboký jehlan. Při tvorbě úlohy je využito vývojové prostředí RobotStudio 4.0, Program-Maker a programovací jazyk Rapid pro roboty ABB.

Celý systém se chová autonomně, kdy jeden z robotů nejprve provede mapování prostoru laserovým snímačem. Ten dále vyhodnotí naměřená data a rozhodne o vlastnostech detekovaného předmětu a pošle je druhému robotu. Ten ve funkci manipulátor po získání informací o poloze a tvaru předmětu provede jeho uchopení a přesunutí na definované místo odkladu. Po zmapování prostoru je tvar a vlastnosti předmětu vykresleny v programu spuštěném na počítači, který načítá data z robota ve funkci snímač. Úloha popsaná v této práci byla úspěšně odzkoušena v laboratoři inteligentních robotů.

Tato práce dále obsahuje popis použitého zařízení, kinematické vztahy související s touto diplomovou prací a dále jsou zde popsány vybrané instrukce jazyka Rapid, které jsou nezbytné pro řešení celé úlohy. Hlavní částí celé práce je podrobně popsaná metoda detekce a rozpoznávání předmětu z hodnot naměřených snímačem MetraLight. Závěrečná část popisuje vyhodnocení funkčnosti celého systému.

## Klíčová slova

Abb, Robot, Kooperace činností, MetraLight, Rapid

# **Abstract**

The aim of this thesis was to creation a functional robotic system for recognition manipulation of objects using a laser sensor. During the entire system control will be all used robotic equipment performing coordinated movements. The system may be used as an alternative to typical industrial excersise detection of object in an area by it's shape, dimensions, orientation and location in the area.

For a practical exercise were used two robots from ABB Company type IRB140 and type IRB1400. For the detection of objects was used a laser sensor MetraLight Rx series, which was attached to the robot arm of IRB 1400 type. Objects used as test examples were in shape of: cylinder, sphere, cone, cube, a regular tetrahedral pyramid, prism with a square at the base, an irregular fourhips pyramid, prism at the base of the rectangle, cylinder lying on it's side, triangular prism, triangular pyramid. For a creation of exercises was used the developing instrument RobotStudio 4.0, ProgramMaker and programming language Rapid.

The entire system works independently, when one of the robots at first map area with a laser sensor. He evaluate the measured data and determine the quality of the detected object and sends it to second robot. The second robot IRB 140 in function of the manipulator after receiving data about the position and shape of the object grips and moves to defined place. Data about detected object are drawn in the program running on a computer, that receives data from the robot in the sensor function. Practical exercises described in this thesis were successfully tested on a real control system in laboratory of the intelligent robots.

The thesis contains used a description of the equipment, kinematic formulas related with this thesis and next there are described chosen instructions of the Rapid language which were necessary for solving sample exercises. The main part of this thesis is in detail described method of detection and recognition of the object from the values measured by the sensor MetraLight. The final section describes an evaluation of the system functionality.

## **Keywords**

ABB, Robot, Coordination of movement, MetraLight, Rapid



# Obsah

Úvod	15
<b>1 Použité zařízení a jeho technická specifikace</b>	<b>17</b>
1.1 Robot s označením IRB 140	17
1.1.1 Pracovní nástroj - pneumatická přísavka	17
1.2 Robot s označením IRB 1400	18
1.2.1 Pracovní nástroj - hlavice pro připevnění snímače MetraLight a pneumatických kleští	19
1.3 Počítač pro vizualizaci dat	20
1.4 Laserový snímač Metralight	21
1.4.1 Vlastnosti snímače	21
1.4.2 Specifikace snímače	21
1.4.3 Komunikace snímače přes rozhraní RS232	21
1.4.4 Sada příkazů pro ovládání snímače	23
1.4.5 Použité módy pro měření objektu	24
<b>2 Kinematika pohybu bodu v prostoru</b>	<b>26</b>
2.1 Popis bodu v prostoru	26
2.2 Výpočet bodu v prostoru	26
2.2.1 Transformační rotační matice pro (3D)	27
2.2.2 Rotace Eulerových úhlů	28
<b>3 Programování v jazyku Rapid</b>	<b>30</b>
3.1 Přerušování v jazyku Rapid	30
3.2 Instrukce pro komunikaci	31
3.2.1 Open	31
3.2.2 ClearIOBuff	32
3.2.3 WriteStrBin	33
3.2.4 ReadStrBin	33
3.3 Instrukce pro konverzi dat	33
3.3.1 StrToVal	34
3.3.2 StrToByte	34
3.4 Instrukce pro získání pozice a natočení externích os robota	34
3.4.1 WorkObject	35
3.4.2 CRobT	36

3.4.3	EulerZYX - (Euler ZYX rotace) . . . . .	36
<b>4</b>	<b>Datová komunikace robotického systému</b>	<b>38</b>
4.0.4	Komunikace mezi snímačem a robotem IRB 1400 . . . . .	38
4.0.5	Komunikace mezi roboty IRB 140 a IRB 1400 . . . . .	39
4.0.6	Komunikace mezi robotem IRB 1400 a PC . . . . .	40
4.0.7	Nastavení sériového rozhraní v řídicím systému S4C+ . . . . .	40
<b>5</b>	<b>Systém rozpoznávání předmětů</b>	<b>42</b>
5.1	Detekce předmětu pomocí přímek . . . . .	43
5.1.1	Definování bodů na snímači . . . . .	43
5.1.2	Princip generování přímek . . . . .	44
5.1.3	Metoda generování přímek - trajektorie pohybu . . . . .	45
5.1.4	Základní klasifikace tvarů předmětů . . . . .	50
5.2	Výpočet průsečíků přímek . . . . .	52
5.3	Shluková analýza - metoda přijatelné vzdálenosti . . . . .	53
5.4	Určení členitosti objektu . . . . .	56
5.4.1	Rozpoznávané předměty - členitost . . . . .	57
5.5	Výpočet rozměrů, orientace a pozice předmětu v prostoru . . . . .	58
5.5.1	Výpočet pro kategorii předmětů 1 . . . . .	58
5.5.2	Výpočet pro kategorii předmětů 2 . . . . .	60
5.5.3	Výpočet pro kategorii předmětů 3 . . . . .	62
5.5.4	Výpočet pro kategorii předmětů 4 . . . . .	65
5.5.5	Předměty nerozpoznávané systémem - kategorie předmětů 5 . . . . .	68
<b>6</b>	<b>Vizualizační aplikace</b>	<b>70</b>
<b>7</b>	<b>Praktické řešení robotické aplikace</b>	<b>71</b>
7.1	Popis jednotlivých činností navrženého systému: . . . . .	71
7.2	Vyhodnocení spolehlivosti, přesnosti, použitelnosti a dosažitelnosti systému . . . . .	74
7.2.1	Pravděpodobnost úspěšného rozpoznání předmětu . . . . .	74
7.2.2	Dosažená přesnost měření celého systému . . . . .	76
7.2.3	Dosažitelnost měřicího systému . . . . .	79
	<b>Závěr</b>	<b>80</b>
	<b>Přílohy</b>	<b>85</b>

## Seznam obrázků

1	Označení rotačních os a pracovního rozsahu IRB 140 . . . . .	17
2	Pneumatická přísavka, vyznačení připevnění k zápěstí robota . . . . .	18
3	Označení rotačních os a pracovního rozsahu IRB 1400 . . . . .	19
4	Nástroj - držák snímače MetraLight a pneumatických kleští, označení připevnění nástroje k zápěstí robota . . . . .	20
5	Laserový snímač Metralight Rx series . . . . .	21
6	Módy dia a), gap b) a center c) . . . . .	24
7	Hranový mód edge1 . . . . .	25
8	Hranový mód edge2 . . . . .	25
9	Bod $B$ pevně spojený se zápěstí robota . . . . .	27
10	Systém RPY . . . . .	29
11	Souřadný systém robota . . . . .	35
12	Schéma propojení celého systému . . . . .	38
13	Umístění portu COM2 a COM3 obrázek a), port COM1 obrázek b) . . . . .	39
14	Schéma propojení pinů dle RS422 . . . . .	39
15	Ikona tlačítka „Další okna“ . . . . .	40
16	Teach Pendant - Položka systémové parametry obr. a), Položka ko- munikace obr. b) . . . . .	41
17	Teach Pendant - Položka výběr komunikačního rozhraní obr. a), Po- ložka nastavení parametrů rozhraní obr. b) . . . . .	41
18	Klasifikátor založený na příznakovém popisu . . . . .	42
19	Konvexní těleso a), Konkávní těleso b) . . . . .	43
20	Umístění bodů: pohled shora . . . . .	44
21	Umístění bodů: boční pohled . . . . .	44
22	Umístění bodů při detekci objektu: pohled shora (režim edge1) . . . . .	44
23	Znázornění detekčních trajektorií . . . . .	46
24	Příklad předmětu s rozdílnými tvary průřezů ve spodní a horní části . . . . .	46
25	Počáteční stav systému . . . . .	47
26	Detekce předmětu v prostoru v ose $x$ . . . . .	47
27	Ohraničení prostoru kde se nachází objekt. Generování přímek $p1$ , $p2$ , $p3$ , $p4$ . . . . .	48
28	Měření výšky předmětu . . . . .	49
29	Generování přímek $p5$ , $p6$ , $p7$ , $p8$ pro řez předmětu . . . . .	49
30	Generování přímek $p9$ , $p10$ , $p11$ , $p12$ pro řez předmětu . . . . .	50

31	Vyznačení výšek při měření průměru předmětu . . . . .	50
32	Tvary detekčních obrazců . . . . .	51
33	Naznačení počítaných průsečíků z přímk . . . . .	52
34	Rozmístění průsečíků při detekci například ve tvaru válce a), kvádru b), krychle c), trojboký hranol d) . . . . .	53
35	Detail shluku tvořeného dvěma průsečíky a vypočítaného centroidu .	54
36	Výpočet centroidu z průsečíků . . . . .	55
37	Znázornění značení členitosti předmětu koule a), kužel b) . . . . .	57
38	Obrázky předmětů z kategorie 1 . . . . .	60
39	Označení stran krychle a), označení stran pravidelného jehlanu b) . .	60
40	Rotace předmětu s podstavou čtverce . . . . .	61
41	Kategorie předmětů 2 . . . . .	62
42	Označení stran kvádru a), označení stran nepravidelného jehlanu b) .	63
43	Rotace předmětu s podstavou obdélníka . . . . .	64
44	Válec položený na bok, znázornění stran pro výpočet . . . . .	65
45	Tvary předmětů kategorie 3 . . . . .	65
46	Nepravidelný trojboký hranol, strany . . . . .	66
47	Rotace předmětu s podstavou trojúhelníka . . . . .	67
48	Kategorie předmětů 4 . . . . .	68
49	Příklad konkávního tělesa s dutinou . . . . .	68
50	Různý tvar průřezu ve spodní a horní části . . . . .	69
51	Naznačení výpočtu podstavy například u komolého a) a nekomolého kuželu b) . . . . .	69
52	Náhled vytvořeného programu pro vykreslení a výpis naměřených dat	70
53	Model laboratoře inteligentních robotů . . . . .	71
54	Vývojový diagram robotického systému . . . . .	73
55	Zobrazení úspěšnosti rozpoznání pro jednotlivé předměty . . . . .	75
56	Přehled grafů zobrazující dosaženou přesnost, výběrový rozptyl a směrodatnou odchylku pro jednotlivé předměty . . . . .	78
57	Naznačení dosažitelnosti ramene robota a vyznačení plochy, na které byla prováděna detekce . . . . .	79

## Seznam tabulek

1	Technická specifikace robota IRB 140 . . . . .	18
2	Technická specifikace robota IRB 1400 . . . . .	19
3	Vlastnosti snímače Rx series . . . . .	22
4	Nastavení pinů RS232 . . . . .	23
5	Transformačních matice pro obecný úhel $\alpha$ v osách x, y, z . . . . .	28
6	Přehledný výpis počítaných průsečíků . . . . .	52

## Seznam zkratek

MTI	Mechatroniky a Technické Informatiky
TCP	Tool Center Point
TP	Teach Pendant
BF	Base Frame
TUL	Technická univerzita v Liberci
CVS	Compact Vision System

# Úvod

Robotika je jedním z nejdůležitějších mezioborových průmyslových odvětví, které zahrnuje znalosti mechatroniky, teorie řízení, měřicí techniky, umělé inteligence a celou řadu dalších vědních disciplín. Robotika je inženýrský obor, ve kterém je kladen veliký důraz na přesnost a spolehlivost systému. Zabývá se návrhem, konstrukcí a vybavením elektromechanického systému, který je schopen vykonávat předem definované a často se opakující činnosti dle požadavků provozovatele. Takovému kompaktnímu systému říkáme robot.

Robot je stroj, který dokáže částečně samostatně vykonávat zadané úkoly. Robot dokáže do určité míry komunikovat se svým okolím, ze kterého získává potřebné informace, podle kterých bude vykonávat své činnosti. Pro komunikaci se svým okolím využívá senzorů. Senzor je vstupním zařízením robota, které dokáže různými způsoby měřit vlastnosti a parametry svého okolí.

Hlavní výhodou robota je jeho možnost opakovaného a rychlého pohybu s velkou přesností, která předčí lidské schopnosti. Díky těmto vlastnostem se stal velice výkonným zařízením, které zjednodušuje pracovní postupy, zrychluje výrobu a celkový výkon jeho provozovatele. Využívání robotů je v moderním průmyslu velice populární, protože svojí samostatností dokážou podniku výrazně zrychlit jeho ekonomický růst.

Z těchto skutečností vyplývají i hlavní cíle a poslání robotiky, která se snaží co nejefektivnějším způsobem využívat moderní technologie pro zjednodušení a zlehčení prováděné práce, kterou by v častých situacích člověk zvládal jen s velikými obtížemi. Robotika je nedílnou součástí evolučního vývoje lidské společnosti.

Téma této diplomové práce bylo vybráno na základě velkého množství uplatnění v průmyslu. Koordinace pohybů robotů se využívá v každé výrobní lince, kde se nachází alespoň dva roboty.

V této práci se omezíme pouze na spolupráci dvou robotů. Oba se nacházejí v laboratoři inteligentních robotů Technické univerzity v Liberci (dále TUL) v učebně S15, budova A. Obě zařízení jsou od mezinárodní švédsko-švýcarské firmy ABB, která se zabývá výrobou robotů už od roku 1972. Roboty od ABB jsou jedněmi z nejrozšířenějších průmyslových robotů na celém světě s nosností do 500 kg. ABB roboty jsou nejčastěji využívány v automobilovém, slévárenském, ale i v potravinářském průmyslu. Tyto roboty mohou využívat přídatných periférií, které je specializují na určitý druh vykonávané činnosti. Příkladem mohou být svářečské soupravy, soupravy na nanášení lepidla a barvy, dále obráběcí frézky případně nástroje pro

manipulaci s předměty nebo také speciální Vision systémy pro rozpoznávání tvarů, rozměrů a barev předmětů. Společnost ABB pro své roboty poskytuje pro snazší ovládání a programování aplikační software nazvaný RobotStudio (dále RS), pomocí kterého je možné nasimulovat offline robotické aplikace.

Tato práce je typickou ukázkou průmyslového využití robotů, které je charakteristické pro robotizovaný systém rozpoznávání a třídění výrobků dle jejich tvaru s určováním pozice, rozměrů a orientace předmětu v prostoru.

První robot bude vykonávat funkci snímač a bude mít ke svému zápěstí připevněnu koncovou hlavici s laserovým snímačem. Pomocí snímače bude detekovat a rozpoznávat předměty. Druhý robot s funkcí manipulátor bude přijímat data od prvního robota, na základně kterých provede manipulační činnost, ve které uchopí nalezený předmět pomocí pneumatické přísavky a přenesení ho na definované místo.

Výsledkem této práce je navrhnout a zrealizovat funkční systém rozpoznávání předmětů pomocí zařízení poskytnutého TUL při vzájemné koordinaci pohybů dvou robotů. První z nich provádí detekční a rozpoznávací činnost pomocí laserového snímače a druhý robot bude provádět manipulační činnost. Po vytvoření bude funkčnost celého systému prakticky ověřena na reálném systému v laboratoři inteligentních robotů v učebně S15, budova A.

## **Práce je rozdělena do několika kapitol**

- V první kapitole se seznámíme s použitým robotickým zařízením, jeho nástroji a laserovým snímačem MetraLight Rx series.
- Druhá část se zaměřuje na kinematické vztahy využité v práci.
- Obsahem třetí kapitoly je popis klíčových instrukcí jazyka Rapid potřebných pro tuto práci.
- Ve čtvrté části se dozvíme, jak správně nastavit všechna komunikační rozhraní potřebná pro posílání dat mezi jednotlivými částmi systému.
- Pátá část popisuje navržené metody pro detekci a rozpoznávání tvarů předmětů, které byly aplikovány.
- V šesté kapitole si stručně popíšeme funkce vytvořeného programu pro výpis a vykreslení naměřených dat.
- V závěrečné sedmé části si popíšeme postupně vykonávané činnosti robotického systému a vyhodnocení přesnosti a úspěšnosti rozpoznávání předmětů.

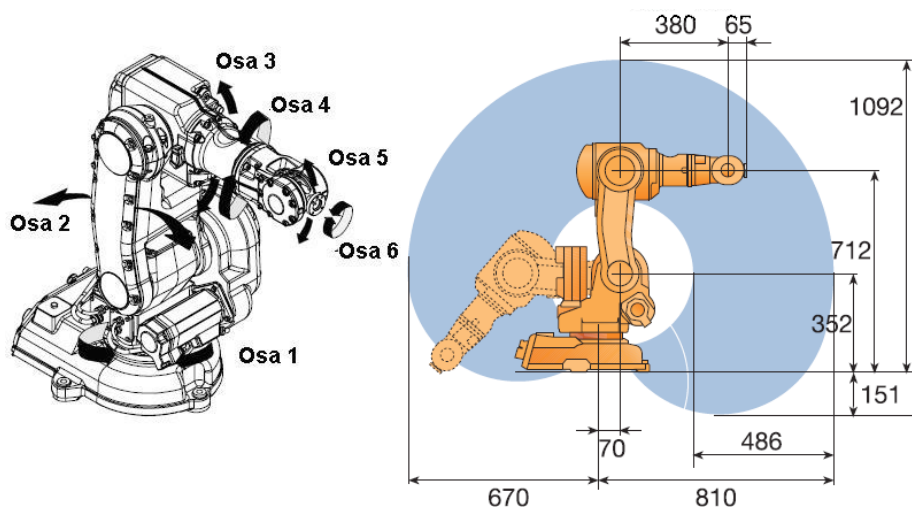


# 1 Použité zařízení a jeho technická specifikace

V první kapitole se seznámíme a vysvětlíme si vše důležité o použitých robotech, laserovém snímači a počítači potřebném k výpisu a vykreslení naměřených dat. Dále u každého z robotů je vyobrazen a popsán pracovní nástroj, který používá, včetně vyznačení připevnění k jeho zápěstí. Vědomosti z této části následně využijeme při realizaci úlohy se snímačem a robotem.

## 1.1 Robot s označením IRB 140

Prvním použitým zařízením je šestiosý robot od firmy ABB s typovým označením IRB 140, který je ve funkci robot-manipulátor. Nachází se ve středu místnosti laboratoře připevněný k desce stolu s řídicím systémem S4C+ postaveným u okna. Jako pracovní nástroj používá pneumatickou přísavku, kterou bude provádět manipulační činnost. Technická specifikace robota (viz tabulka 1) a dále vyznačení rotačních os a jeho dosahu ramene (viz obrázek 1). Více informací o použitém robotu IRB 140 je možné získat z „Product Manual IRB140.pdf“ přiloženém na CD k této práci.



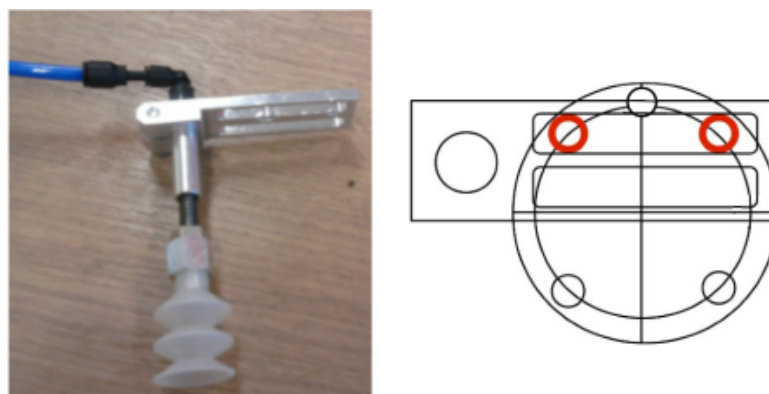
Obrázek 1: Označení rotačních os a pracovního rozsahu IRB 140

### 1.1.1 Pracovní nástroj - pneumatická přísavka

Pracovní nástroj pneumatická přísavka je připevněna dvěma šrouby k zápěstí robota (viz obrázek 2). Tímto nástrojem bude robot IRB 140 provádět manipulační činnosti s rozpoznávanými předměty.

Hmotnost robota	95 kg
Počet os volnosti	6
Dosah ramene	810 mm
Přesnost pohybu	0.05 mm
Max. hmotnost pracovního nástroje	5 kg
Max. rychlost pohybu	2,1 m/s
Počet signálů na horním rameni	12
Datová komunikace	RS232, RS422, Ethernet

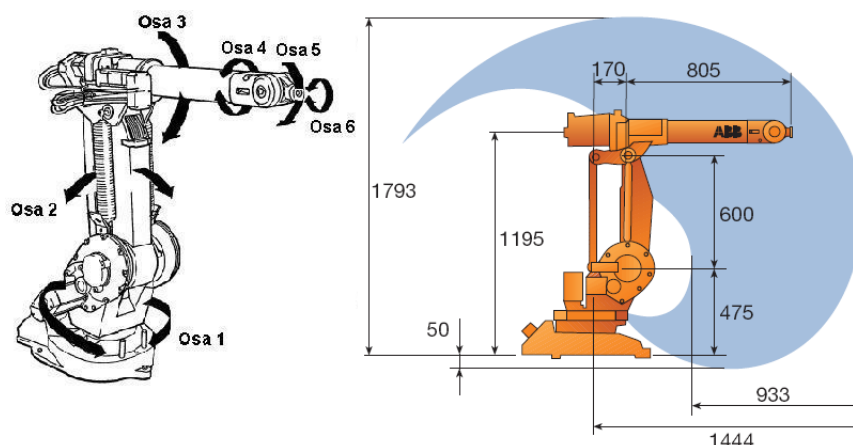
Tabulka 1: Technická specifikace robota IRB 140



Obrázek 2: Pneumatická přísavka, vyznačení připevnění k zápěstí robota

## 1.2 Robot s označením IRB 1400

Druhým použitým robotickým zařízením v této práci je opět robot od firmy ABB s typovým označením IRB 1400, který je ve funkci robot-snímač. Robot se nachází u vstupních dveří do laboratoře, kde je umístěn i jeho řídicí systém S4C+. Tento robot bude využívat nástroje, který je kombinací laserového snímače MetraLight a pneumatických kleští připevněných k zápěstí robota. Robot bude sloužit pro detekci a rozpoznávání předmětu v prostoru pomocí snímače MetraLight. Technická specifikace robota (viz tabulka 2) a dále označení rotačních os a rozsah pracovního prostoru (viz obrázek 3). Více informací o použitém robotu IRB 1400 je možné získat z „Product Manual IRB1400.pdf“ přiloženém na CD k této práci.



Obrázek 3: Označení rotačních os a pracovního rozsahu IRB 1400

Hmotnost robota	225 kg
Počet os volnosti	6
Dosah ramene	1440 mm
Přesnost pohybu	0.05 mm
Max. hmotnost pracovního nástroje	5 kg
Max. rychlost pohybu	2,1 m/s
Počet signálů na horním rameni	12
Datová komunikace	RS232, RS422, Ethernet

Tabulka 2: Technická specifikace robota IRB 1400

### 1.2.1 Pracovní nástroj - hlavice pro připevnění snímače MetraLight a pneumatických kleští

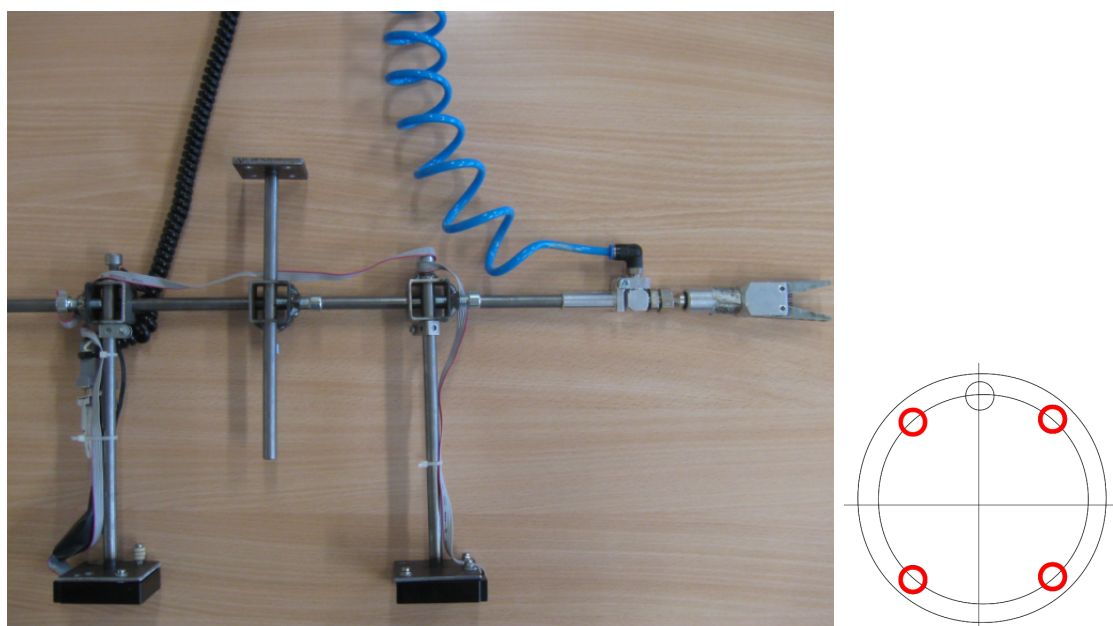
Tento nástroj byl vytvořen za účelem vhodného přichycení laserového snímače MetraLight k rameni robota. Nástroj dále obsahuje výstup na připevnění pneumatických kleští s možností nahrazení za pneumatickou přísavku pro manipulační činnost. Koncová hlavice je uchycena čtveřicí šroubů k zápěstí robota. Celý nástroj je na obrázku, včetně vyznačení jeho připevnění k zápěstí robota (viz 4). Více o snímači MetraLight (viz část 1.4). Tento nástroj byl vytvořen po konzultaci s vedoucím práce a Ing. Janem Pluhařem. Rozpětí konstrukce je 230 mm, měřeno mezi oběma částmi snímače.

Koncová hlavice robota je složena z několika dílů, díky kterým je rozkládací a

variabilně nastavitelná dle potřeby.

**Celá konstrukce se skládá:**

- držák konstrukce spojující zápěstí robota s nástrojem
- ze tří jeklů, každý se dvěma závity pro upevnění konstrukce
- z jedné tyče spojující celou konstrukci s uchycenými pneumatickými kleštěmi
- ze dvou držáků, každý z nich pro jednu část snímače MetraLight (receiver, emitter)



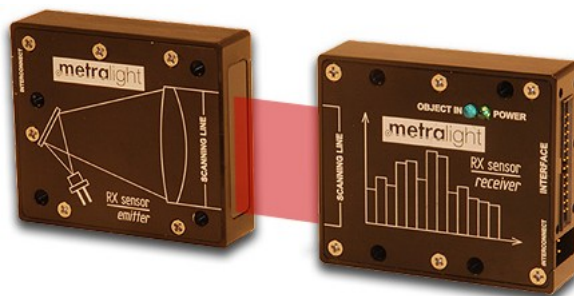
Obrázek 4: Nástroj - držák snímače MetraLight a pneumatických kleští, označení připevnění nástroje k zápěstí robota

### 1.3 Počítač po vizualizaci dat

Tento počítač (dále už jen PC) byl přidán po konzultaci s vedoucím práce s ohledem na lepší a průkaznější podání naměřených hodnot obsluze systému. Pro zobrazování naměřených hodnot byl vytvořen program (viz kapitola 6). Jako PC může být použit jakýkoliv, ale je zapotřebí, aby měl sériový port RS232 pro požadovanou komunikaci s řídicím systémem robota.

## 1.4 Laserový snímač Metralight

V této části si přiblížíme vlastnosti a technickou specifikaci použitého snímače Metralight Rx series (dále snímač) a jeho parametry. Více se také seznámíme s použitými měřicími módy a také se základními příkazy pro jeho ovládání.



Obrázek 5: Laserový snímač Metralight Rx series

### 1.4.1 Vlastnosti snímače

Jedná se o bezkontaktní laserový snímač kompaktní velikosti, který je schopný pořizovat a zpracovávat data v reálném čase. Je vyroben s důrazem na rychlé a přesné měření. Zařízení se skládá ze dvou částí, vysílače (emitor) a přijímače (receiver), které jsou propojeny datovým kabelem pro vzájemnou komunikaci. Snímač dokáže eliminovat okolní zdroje světla a měří hned několik různých parametrů: hrany, mezery, velikost objektu a další.

Pro připojení je možné využít několik různých rozhraní: Paralelní, Sériové, Analogové, USB nebo ethernetové rozhraní. Snímač byl dodán TUL se sériovým rozhraním RS232, které bylo využito pro komunikaci.

### 1.4.2 Specifikace snímače

Nyní se podíváme na všechny důležité parametry snímače přehledně seřazené v tabulce č.3.

### 1.4.3 Komunikace snímače přes rozhraní RS232

V této části je uvedeno vnitřní nastavení komunikace snímače přes sériový port RS232. Rozhraní RS232 je výhodné z hlediska připojitelnosti snímače k řídicímu systému robota, který má dva takovéto porty s maximální přenosovou rychlostí 38400 b/s.

Metoda detekce	Laser procházející čočkou s CCD snímačem
Zdroj světla	Laserová dioda, 780nm
Rozsah měření	140 micronů až 28 mm (0.006 - 1.10 palců)
Rozlišení snímače	7 micronů (RX07), 3,5 micronů (RX03)
Opakovatelnost	14 micronů
Doba odezvy	0,417 ms
Nelinearita	0,3% z plného rozsahu
Napájení	12 až 24 VDC / 170 mA nebo přes USB
Komunikační rozhraní	Paralelní port (TTL), sériový port (RS232), analogový výstup nebo USB
Indikátory	Zelená LED = snímač zapnutý Modrá LED = je přítomný objekt
Formát dat	12/13 bitový paralelní, ASCII řetězec nebo analogové napětí
Fyzické rozměry	Vysílač i přijímač shodně: 584 mm x 508 mm x 152 mm

Tabulka 3: Vlastnosti snímače Rx series

### Parametry komunikace snímače

- Přenosová rychlost: standardně 115200 b/s, v této práci je použito 38400 b/s.
- Datové bity: 8
- Parita: žádný
- Řízení toku: žádný
- Nastavení ASCII: zakončení řádků kódem odřádkování

V následující tabulce č. 4 je vypsáno připojení jednotlivých pinů pro komunikaci po sériové lince. Všechny ostatní piny jsou neobsazené nebo nevýznamné pro rozhraní RS232.

Pin	Typ	Jméno	Popis
1	OUT	TXD	Odesílání dat
2	IN	RXD	Příjem dat
3	GND	GND	Zem
21,22	PWR	+PWR	+12 až +24 VDC
23,24	GND	GND	Zem

Tabulka 4: Nastavení pinů RS232

#### 1.4.4 Sada příkazů pro ovládání snímače

Sada příkazů, která umožňuje práci se snímačem s verzí firmware v1.f0, v1.g0 a v1.h0. Jsou k dispozici příkazy pro získání informací o měření ze snímače, ale také nastavení různých parametrů nebo módů snímání. Podrobně si ukážeme příkazy v následujícím přehledu.

##### 1. d<CR><LF>

- **Popis:** vrací naměřenou hodnotu. Pokud je nastaven počet vzorků, (viz příkaz č.3) pro výpočet hodnoty je použit jejich průměr.
- **Odpověď:** d<CR><LF>hodnota<CR><LF>
- **Příklad:** Pokud snímač vrátí hodnotu 1F40, musíme jí převést do desítkové soustavy,  $1F40_{16} = 8000_{10}$ , tím získáme počet bodů, který vynásobením hodnotou 0,0035 převedeme na mm.  $8000 \cdot 0,0035 = 28\text{mm}$

##### 2. o<CR><LF>

- **Popis:** zobrazí status\_objektu - objekt přítomen / nepřítomen
- **Odpověď:** o<CR><LF>status\_objektu<CR><LF>  
status\_objektu = 00000 (nepřítomen) nebo 00001 (přítomen)

##### 3. a[pocet\_vzorku]<CR><LF>;

00001<pocet\_vzorku<00127

- **Popis:** nastaví počet vzorků pro výpočet průměrné hodnoty. Pokud není parametr pocet\_vzorku zadán, zobrazí se aktuální nastavená hodnota.
- **Odpověď:** a<CR><LF>pocet\_vzorku<CR><LF>

4. **m**[**cislo\_modu**]<CR><LF>;

0<cislo\_modu<7

- **Popis:** nastaví mód měření. Pokud není parametr **cislo\_modu** zadán zobrazí se aktuálně nastavený mód měření.

m0=edge1, m1=edge2, m2=diameter, m3=gap, m4=center,  
m5=firstedge, m6=firstdiameter, m7=custom

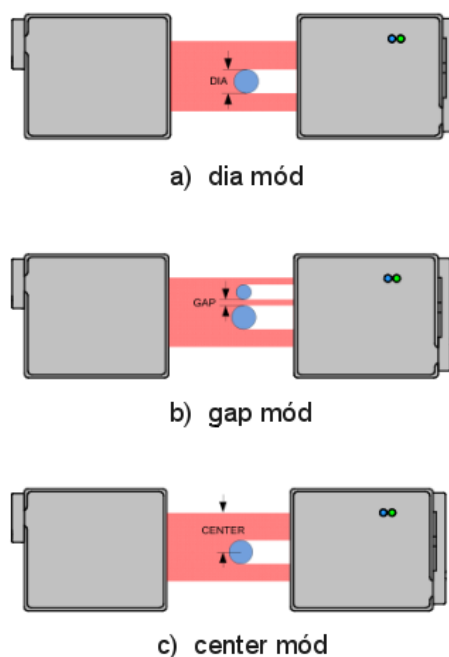
- **Odpověď:** m<CR><LF>cislo\_modu<CR><LF>

5. **v**<CR><LF>

- **Popis:** zobrazí verzi firmwaru snímače
- **Odpověď:** verze\_firmware<CR><LF>

#### 1.4.5 Použité módy pro měření objektu

Snímačem lze měřit různými způsoby parametry měřeného objektu. Například velikost průměru předmětu (dia mód), vstupní hrany (edge1 mód), výstupní hrany (edge2 mód), vzdálenost mezi dvěma předměty (gap mód), pozice středu objektu (center mód), atd.



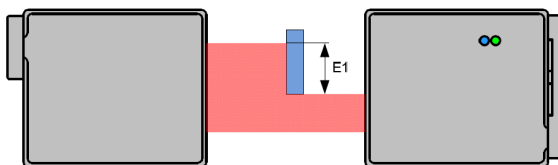
Obrázek 6: Módy dia a), gap b) a center c)

Pro snazší využití snímače by pro naši aplikaci bylo výhodné, kdyby se celý předmět vešel do měřicího rozsahu snímače. To by znamenalo vzhledem k technickým



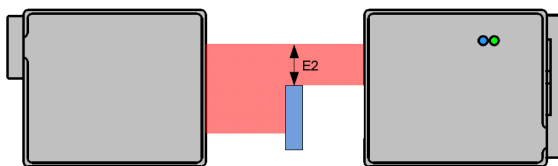
parametry snímače měřit malé předměty o maximální šíři 28mm na jeho nejdelší hraně nebo mít větší měřicí plochu snímače. První možnost je dost omezující a druhá je bohužel nemožná, jelikož není k dispozici snímač s větším rozsahem. Z toho vyplývá, že bude potřeba vymyslet vhodnou kombinaci dostupných módů snímače a pohybu robota. Jako nejvhodnější pro náš způsob měření vycházejí hranové módy, pomocí kterých můžeme měřit předměty vstupující a vystupující z rozsahu snímače. Módy použité pro měření objektu:

- Mód edge1 (hranový mód)
  - tento mód měří velikost hrany vstupující do měřeného rozsahu snímače (viz obrázek 7). Přepnutí snímače do tohoto módu se provede příkazem [m0] (viz část 1.4.4).



Obrázek 7: Hranový mód edge1

- Mód edge2 (hranový mód)
  - tento mód měří velikost hrany vystupující z měřeného rozsahu snímače (viz obrázek 8). Přepnutí snímače do tohoto módu se provede příkazem [m1] (viz část 1.4.4).



Obrázek 8: Hranový mód edge2

## 2 Kinematika pohybu bodu v prostoru

Tato kapitola byla zařazena do práce z důvodu využití potřebných kinematických vztahů pro řešení zadané úlohy.

Kinematika se zabývá pohybem hmotného bodu v rovině nebo prostoru. Hmotným bodem je v kinematice nahrazováno skutečné pevné těleso, jehož tvar a rozměry nejsou pro sledovaný jev podstatné a lze je tedy zanedbat. Prostorovým pohybem budeme rozumět pohyb pevného tělesa v prostoru.

Pro využití v této práci se budeme zabývat prostorem  $R^3$ . Při definované souřadné soustavě a pro nás známém pohybu tělesa v něm, je pak možné dopočítat polohu bodu s tímto tělesem spojeným. Pro naše účely se bude jednat o body určené na obou částech snímače, které definují přímky kolem detekovaného tělesa v prostoru (viz kapitola 5). Oba body jsou vztaženy ke koncovému bodu nástroje Tool Center Point (dále už jen TCP). Pozici a natočení TCP budeme vždy v danou chvíli znát díky instrukci CRobT (viz 3.4.2). Velikost změny pozice bodů na snímači je dána velikostí úhlu natočení TCP a jeho pozice ve všech třech osách (x,y,z).

### 2.1 Popis bodu v prostoru

Každý bod prostoru má přesně definovanou polohu od středu souřadného systému, ke kterému je vztažený. Dále pro každý bod budeme určovat výslednou orientaci nástroje při jeho dosažení.

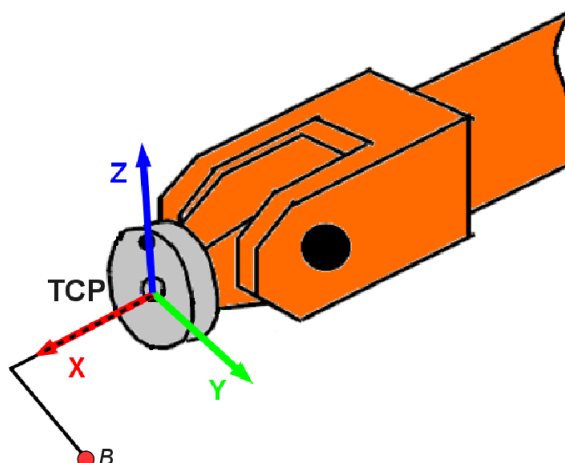
- **Poloha bodu** - je hodnota, vyjadřující pozici bodu v prostoru vzhledem k aktuálnímu souřadnému systému, od kterého je vzdálený v osách x, y, z.
- **Orientace nástroje** - je určena velikostí natočení nástroje při dosažení cílového bodu v úhlech  $(\alpha, \beta, \gamma)$  v osách x, y, z od počátečního natočení souřadného systému.

Pro naše využití je středem souřadného systému WorkObject (viz část 3.4.1).

### 2.2 Výpočet bodu v prostoru

V této části se budeme zabývat výpočtem nové pozice bodu v prostoru, který je spojený s TCP robota. Tento spojený bod se bude posouvat a rotovat stejným způsobem jako TCP. Pro názornost definujeme obecný bod pevně spojený s TCP, označen písmenem  $B$  (viz obrázek 9). Všechny nové pozice bodu  $B$  jsou vypočítávány vzhledem k pevnému souřadnému systému robota. Z pozice a orientace TCP v každém bodu

trajektorie budeme počítat novou polohu bodu  $B$  v prostoru.



Obrázek 9: Bod  $B$  pevně spojený se zápěstí robota

Ke zjištění pozice bodu  $B$  v prostoru jsou uvedeny dva matematické aparáty:

- Transformační rotační matice
- Rotace Eulerových úhlů

Existují i jiné metody výpočtu, ale pro práci se zaměříme na již zmíněné, které naprosto vyhovují našemu použití.

### 2.2.1 Transformační rotační matice pro (3D)

První možností, jak vypočítat pozici bodu  $B$  v prostoru při známém natočení TCP o obecný úhel  $\alpha$  kolem osy rotace, je Transformační rotační matice. Polohu TCP v prostoru získáme pomocí funkce CRobT (viz 3.4.2). Orientaci TCP ( $\alpha$ ,  $\beta$ ,  $\gamma$ ) v cílovém bodě získáme pomocí funkce EulerZYX (viz 3.4.3). Pro výpočet v 3D prostoru se používají transformační rotační matice pro rotaci kolem všech tří os (viz tabulka 2.2.1).

#### **Příklad:**

Mějme bod  $B = (x_1, y_1, z_1)$  pevně spojený s TCP robota. Vypočtěme novou polohu bodu  $B = (x_2, y_2, z_2)$  pro rotaci TCP o úhel  $\alpha$  kolem osy  $z$ . (Použijte pravotočivé otáčení)

#### **Aplikaci rotační matice provedeme následovně:**

Rotační matice pro rotaci kolem osy  $z$  při natočení o úhel  $\alpha$  proti směru pohybu otáčení hodinových ručiček.

$$\mathbf{R}(\alpha, \mathbf{z}) = \begin{pmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (1)$$

Daná matice pro rotaci v ose  $z$  je násobena velikostí polohového vektoru bodu  $B$  od robotova TCP.

$$\begin{pmatrix} x_2 \\ y_2 \\ z_2 \end{pmatrix} = \begin{pmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix} \quad (2)$$

Vypočítaná nová pozice bodu  $B$  po natočení TCP o úhel  $\alpha$  při pravotočivém otáčení má polohu  $B = (x_2, y_2, z_2)$ .

Osa otáčení je rovnoběžná s osou	X			Y			Z		
Pravotočivé otáčení bodu	1	0	0	$\cos \alpha$	0	$\sin \alpha$	$\cos \alpha$	$-\sin \alpha$	0
	0	$\cos \alpha$	$-\sin \alpha$	0	1	0	$\sin \alpha$	$\cos \alpha$	0
	0	$\sin \alpha$	$\cos \alpha$	$-\sin \alpha$	0	$\cos \alpha$	0	0	1
Levotočivé otáčení bodu	1	0	0	$\cos \alpha$	0	$-\sin \alpha$	$\cos \alpha$	$\sin \alpha$	0
	0	$\cos \alpha$	$\sin \alpha$	0	1	0	$-\sin \alpha$	$\cos \alpha$	0
	0	$-\sin \alpha$	$\cos \alpha$	$\sin \alpha$	0	$\cos \alpha$	0	0	1

Tabulka 5: Transformačních matice pro obecný úhel  $\alpha$  v osách  $x, y, z$

### 2.2.2 Rotace Eulerových úhlů

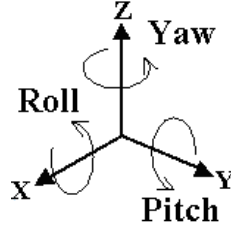
Druhou možností, jak vypočítat polohu bodu  $B$  v prostoru, je pomocí Eulerových úhlů. Jedná se o soustavu tří úhlů  $(\psi, \vartheta, \varphi)$  při postupném pootočení souřadnicových os v daných souřadných rovinách. Jednotlivé úhly  $(\psi, \vartheta, \varphi)$  náležejí těmto osám rotace  $\rightarrow (z, y, x)$ . Velikosti jednotlivých úhlů získáme z funkce EulerZYX (viz 3.4.3). Během rotace je nutné dodržovat pořadí natáčení os, okolo kterých otáčíme bod. Například různé výsledky získáme pro otáčení XZY, než pro jiné uspořádání. Pro naše použití budeme otáčet ZYX.

**Takto vyjádříme matematicky postupné otáčení:**

$$R_{RPY} = R_Z \cdot R_Y \cdot R_X \quad (3)$$

### Z angličtiny zkratka RPY:

R - roll, P - pitch, Y - yaw. V češtině se používá „otáčení“, „klopení“ a „bočení“.



Obrázek 10: Systém RPY

Matice  $R_{RPY}$  pro postupné natáčení ZYX vypadá takto:

$$\mathbf{R}_{RPY} = \begin{pmatrix} \cos\vartheta \cdot \cos\psi & -\sin\psi \cdot \cos\varphi + \cos\psi \cdot \sin\vartheta \cdot \sin\varphi & \sin\psi \cdot \sin\varphi + \cos\psi \cdot \sin\vartheta \cdot \cos\varphi \\ \sin\psi \cdot \cos\vartheta & \cos\psi \cdot \cos\varphi + \sin\psi \cdot \sin\vartheta \cdot \sin\varphi & -\cos\psi \cdot \sin\varphi + \sin\psi \cdot \sin\vartheta \cdot \cos\varphi \\ -\sin\vartheta & \cos\vartheta \cdot \sin\varphi & \cos\varphi \cdot \cos\vartheta \end{pmatrix} \quad (4)$$

### Příklad:

Mějme bod  $B = (x_1, y_1, z_1)$  pevně spojený s TCP. Vypočtíme novou polohu bodu  $B = (x_2, y_2, z_2)$  při rotaci TCP o úhel  $\psi, \vartheta, \varphi$ .

**Aplikace Eulerových úhlů pro obecné natočení v systému:**

$$\begin{pmatrix} x_2 \\ y_2 \\ z_2 \end{pmatrix} = \begin{pmatrix} \cos\vartheta \cdot \cos\psi & -\sin\psi \cdot \cos\varphi + \cos\psi \cdot \sin\vartheta \cdot \sin\varphi & \sin\psi \cdot \sin\varphi + \cos\psi \cdot \sin\vartheta \cdot \cos\varphi \\ \sin\psi \cdot \cos\vartheta & \cos\psi \cdot \cos\varphi + \sin\psi \cdot \sin\vartheta \cdot \sin\varphi & -\cos\psi \cdot \sin\varphi + \sin\psi \cdot \sin\vartheta \cdot \cos\varphi \\ -\sin\vartheta & \cos\vartheta \cdot \sin\varphi & \cos\varphi \cdot \cos\vartheta \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix} \quad (5)$$

Po vynásobení matice polohovým vektorem bodu  $B = (x_1, y_1, z_1)$  od robotova TCP, získáme novou polohu bodu  $B$  po pootočení TCP o obecné úhly  $(\psi, \vartheta, \varphi)$ .

## 3 Programování v jazyku Rapid

Tato část pojednává především o popisu a vysvětlení složitějších příkazů programovacího jazyka Rapid, které je potřeba znát pro praktické vytvoření zadané úlohy.

Programovací jazyk Rapid je souborem instrukcí pro ovládání robotů ABB. Instrukce jsou pro všechny roboty ABB stejné, což je velice příjemné, jelikož se dají programovat od nejmenších, až po největší stále stejným jazykem. Syntaxe jazyka je velice podobná programovacímu jazyku Pascal a proto je velice jednoduché se naučit jeho základní příkazy. Pro vytvoření zdrojového kódu robota postačí i jednoduchý textový editor. Pro větší programátorský komfort zde existuje přímo od ABB program pro tvorbu a editaci kódu, jenž se jmenuje ProgramMaker, který obsahuje i kontrolu napsaného kódu. Všechny vytvořené programy pro ABB roboty musí mít koncovku \*.prg. Více informací o programování ABB robotů naleznete na přiloženém CD v „Rapid reference Manual.pdf“.

### 3.1 Přerušování v jazyku Rapid

V jazyku Rapid není možné spouštět paralelně více instrukcí a je tedy nutné si poradit jiným způsobem. Například i při vykonávání instrukce pohybu robota nelze vykonávat další instrukce bez použití přerušování. Jako jediné možné řešení se jeví použití funkce přerušování pomocí instrukce ITimer. Oproti nativní podpoře vícevláknového běhu programu je hlavní nevýhoda v nízké frekvenci vykonávání přerušování. Přerušování je možné pustit každých 0.05s, ale pro cyklické přerušování je vhodné nepoužívat časové kvantum menší jak 0.25s pro řídicí systémy S4C+. Po vyvolání přerušování se spustí rutina obsluhy přerušování (trap rutina), ve které jsou námi zadané instrukce.

Pro naše účely bylo nutné použít přerušování při zjišťování pozice objektu. Robot je při skenování v pohybu a proto bylo přerušování jediné možné řešení. Během cyklického přerušování je navazována komunikace se snímačem a testuje se přítomnost objektu v skenované oblasti.

**Příklad:**

```
VAR intnum timeint;           !deklarace promenne casoveho kvanta
...
IEnable;                      !aktivace preruseni
CONNECT timeint WITH detect;  !pripojeni preruseni, po casovem
                              kvantu spusteni trap rutiny detect
ITimer 0.25, timeint;         !hodiny, casove kvantum 0.25s
```

```

do promenne timeint
...
IDisable;           !deaktivace preruseni

```

V tomto příkladu nejprve aktivujeme přerušeni. Poté systém informujeme o tom, ke které rutině se má připojit, když přerušeni nastane. V našem případě se spustí trap rutina „detect“. Do ITimer nastavujeme 0.25, tj. přerušeni se spustí po každých 0.25 sekundách a v něm se vykoná rutina „detect“. Nakonec deaktivujeme přerušeni.

### Deklarace trap rutiny:

```

TRAP detect
...příkazy které se provedou během přerušeni...
ENDTRAP

```

## 3.2 Instrukce pro komunikaci

Pro komunikaci mezi snímačem MetraLight a řídicím systémem robota je použito sériového rozhraní. V sekci věnované snímači jsme se dozvěděli, jaké znaky mu posílat pro získání potřebných informací. Nyní je potřeba popsat způsob, jak tyto znaky zapisovat a číst ze sériového kanálu pomocí jazyku Rapid.

### 3.2.1 Open

Instrukce Open se používá pro otevírání sériového kanálu, nebo souboru pro čtení a zápis.

**Syntaxe:** Open Objekt \File, IODevice, \Read, \Write, \Append, \Bin;

- *Object* - I/O objekt (I/O zařízení), které má být otevřeno například, „HOME:“, „flp1:“, „com2:“ nebo „pc“.
- *\File* - jméno souboru, který má být otevřen.
- *IODevice* - odkaz k otevření souboru nebo sériového kanálu. Tento odkaz je pak užívaný pro čtení a zápis ze nebo do souboru či sériového kanálu.
- *\Read* - otevře soubor nebo sériový kanál pro čtení. Čtení ze souboru vždy začíná na začátku souboru.

- `\Write` - otevře soubor nebo sériový kanál pro zápis. Při výběru existujícího souboru, bude obsah smazán a přepsán novým záznamem. Vždy se zapisuje na začátek souboru.
- `\Append` - otevírá soubor nebo sériový kanál pro zápis. Když vybraný soubor existuje, tak cokoliv připsané do souboru bude zapsáno na konec.

Argumenty `\Read`, `\Write`, `\Append` se vzájemně vylučují. Když žádný z nich není specifikovaný, instrukce jedná stejným způsobem jako `\Write` argument pro znakové soubory nebo sériový kanál (bez instrukce `\Bin` argumentu) a stejným způsobem jako `\Append` argument pro binární soubory nebo sériový kanál (s instrukcí `\Bin` argumentu). `\Bin` - soubor nebo sériový kanál je otevřený pro binární mód. Jestliže nejsou zadány argumenty `\Read`, `\Write` nebo `\Append`, pak instrukce `Open` otevře binární soubor nebo sériový kanál pro čtení a zápis s ukazatelem na konci souboru.

#### **Příklad:**

```
Var iodev chan;
...
Open "com2", chan\Bin;
```

Otevře sériový kanál na portu com2, který se jmenuje „chan“. Příkazem `Close` se sériový kanál uzavírá (deaktivuje).

### **3.2.2 ClearIOBuff**

Používá se pro mazání dat ve vyrovnávací paměti (buffer) sériového kanálu.

**Syntaxe:** `ClearIOBuff IODevice;`

- *IODevice* - jméno používaného sériového kanálu.

#### **Příklad:**

```
VAR iodev chan;
...
Open "com2", chan \Bin;
ClearIOBuff chan;
```

V tomto příkladě jsme provedli vymazání vyrovnávací paměti pro sériový kanál „chan“.



### 3.2.3 WriteStrBin

WriteStrBin se používá pro zápis řetězce znaků (string) do binárního sériového kanálu nebo binárního souboru. Tuto instrukci využíváme při ovládání snímače skrze sériový kanál, který je otevřený v binárním módu.

**Syntaxe:** WriteStrBin ([\IODevice] [\Str]);

- *IODevice* - jméno používaného sériového kanálu.
- *Str* - zapsaný řetězec do sériového kanálu

**Příklad:**

```
Var iodev ComChannel;  
WriteStrBin ComChannel,"o\0D\0A";
```

Provede se zápis instrukce „o“ do sériového kanálu. Příkazy CR (Carriage Return) a LF (Line Feed) nahradíme jejich hexa hodnotou z ASCII tabulky.

CR = "0D", LF = "0A"

### 3.2.4 ReadStrBin

Je instrukce, která se používá pro čtení řetězce znaků (string) ze sériového kanálu. Pro správné načtení dat ze snímače musí být sériový kanál otevřen v binárním módu. Snímač vrací požadovaná data ve tvaru string, okamžitě po získání dotazu z řídicího systému.

**Syntaxe:** WriteStrBin ([\IODevice] [\NoOfChars] [\Time]);

- *IODevice* - jméno používaného sériového kanálu.
- *NoOfChars* - počet přečtených znaků ze sériového kanálu
- *Time* - délka načítání dat ze sériového kanálu v sekundách

**Příklad:**

```
Var iodev ComChannel;  
ReadStrBin(ComChannel,5\Time:=5);
```

Načte ze sériového kanálu prvních 5 znaků, čeká 5s.

## 3.3 Instrukce pro konverzi dat

Důležité instrukce pro převody textových formátů.

### 3.3.1 StrToVal

Instrukce převádějící hodnotu z řetězce znaku (string) obsahující numerické znaky do číselného proměnného typu num.

**Syntaxe:** StrToVal ([\Str] [\Val]);

- *Str* - řetězec znaků typu string obsahující znaky čísel.
- *Val* - cílová proměnná typu Num.

**Příklad:**

```
VAR String PosY:="689";  
VAR num IsIn;  
VAR bool ok;  
ok:=StrToVal(PosY,IsIn);
```

Příklad převede řetězec "689" do číselné podoby a uloží do proměnné IsIn. V proměnné ok se uloží hodnota „True“ pokud se konverze provede správně, jinak hodnota „False“.

### 3.3.2 StrToByte

Jedná se o instrukci, která převádí řetězce znaků (string) do číselné podoby typu Byte. Tvar řetězce znaků může být následující: hexadecimální (\Hex), osmičkový (\Okt), binární (\Bin) nebo znakový (\Char). Množina přijatelných znaků pro konvertování je (0-9, A-F).

**Syntaxe:** StrToVal ([\Hex] [\Okt] [\Bin] [\Char]);

**Příklad:**

```
VAR string PosHex:="AB";  
VAR num vysledek;  
vysledek:=StrToByte(PosHex\Hex);
```

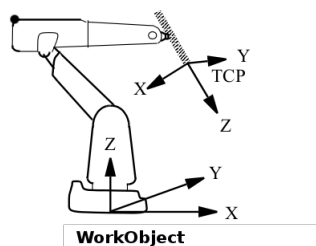
Tento příklad převede hexadecimální hodnotu AB do desítkové soustavy (171).

## 3.4 Instrukce pro získání pozice a natočení externích os robota

Obsahuje popis instrukce definice počátku souřadného systému robota a dále vysvětlení instrukcí pro načítání pozice a natočení koncového nástroje robota.

### 3.4.1 WorkObject

WorkObject (dále WObj) je důležitou součástí každého programu robotické aplikace. WObj určuje počátek souřadného systému, od kterého se počítají souřadnice TCP robota. Při posunu WObj se posouvá i celý souřadný systém a s ním i všechny body k němu vztažené. Při nedefinování vlastního WObj v programu se vygeneruje základní s názvem wobj0 s pozicí ( $x=0$ ,  $y=0$ ,  $z=0$ ) v místě paty robota nazývaném Base Frame (BF).



Obrázek 11: Souřadný systém robota

#### Popis složek WObj:

- *robhold* - definuje, zda robot drží WObj.
  - TRUE - robot drží WObj (WObj definovaný na TCP robota), nastaví se při použití statického nástroje (nástroj není připevněn k zápěstí).
  - FALSE - robot nedrží WObj, nastaví se při definovaném nástroji robota (připevněného k zápěstí).
- *ufprog* - definuje, zda je nebo není použitý pevný uživatelský systém.
  - TRUE - je použit pevný uživatelský systém.
  - FALSE - je použit pohyblivý uživatelský souřadný systém.
- *ufmec* - definuje pohybovou jednotku, se kterou robot provádí koordinované pohyby, například posuvný pás. Je možné ho nastavit, má-li *ufprog* hodnotu FALSE.
- *uframe* - určuje pozici a natočení právě aktuálního WObj v uživatelském souřadném systému. Pozice je dána v osách ( $x$ ,  $y$ ,  $z$ ) a natočení v kvaternionech ( $q_1$ ,  $q_2$ ,  $q_3$ ,  $q_4$ ).
- *oframe* - určuje pozici a natočení právě aktuálního WObj v objektovém souřadném systému. Pozice je dána v osách ( $x$ ,  $y$ ,  $z$ ) a natočení v kvaternionech ( $q_1$ ,  $q_2$ ,  $q_3$ ,  $q_4$ ).

### Příklad:

```
PERS wobjdata MyWObj := [FALSE, TRUE, "", [[0, 100, 0],  
[1, 0, 0, 0]], [[0, 0, 0], [1, 0, 0, 0]]];
```

V příkladě umístíme MyWObj na pozici (x=0, y=100, z=0) s nulovou rotací.

### 3.4.2 CRobT

Načte aktuální pozici koncového bodu robota. Tato funkce vrací hodnotu robtarget s pozicí (x, y, z), orientaci (q1, q2, q3, q4), vzhledem k použitému WObj .

**Syntaxe:** CRobT ([\Tool] [\WObj]);

- *Tool*: nástroj, pro který se počítá koncová pozice robota. Pokud není uvedeno, je použit aktuální nástroj.
- *WObj*: WorkObject, ke kterému je pozice vypočítána.

### Příklad:

```
Var robtarget pos;  
pos:=CRobT(\MyTool, \MyWobj);
```

Uloží aktuální polohu a natočení nástroje MyTool vzhledem k WorkObjectu MyWobj.

### 3.4.3 EulerZYX - (Euler ZYX rotace)

Načítá velikosti eulerových úhlů (z, y, x) z orientace objektu vzhledem k počátku soustavy. Hodnoty orientace objektu jsou v quaternionech. Jde tedy o funkci, která převede quaterniony na velikosti úhlů v osách rotace (x, y, z). Odpovídající Eulerův úhel, vyjádřen ve stupních, je v rozsahu [- 180, + 180]. Vrací hodnotu (num).

**Syntaxe:** EulerZYX ([\X] [\Y] [\Z] ObjektRotace);

- $\backslash X$  - vrací úhel v ose rotace X vzhledem k počátku soustavy.
- $\backslash Y$  - vrací úhel v ose rotace Y vzhledem k počátku soustavy.
- $\backslash Z$  - vrací úhel v ose rotace Z vzhledem k počátku soustavy.

**Příklad:**

```
VAR num xuhel;  
VAR num yuhel;  
VAR num zuhel;  
VAR pose object;  
...  
xuhel := EulerZYX(\X, object.rot);  
yuhel := EulerZYX(\Y, object.rot);  
zuhel := EulerZYX(\Z, object.rot);
```

Výsledkem tohoto příkladu jsou uložené úhly natočení objektu v osách (x, y, z) do proměnných xuhel, yuhel, zuhel.

## 4 Datová komunikace robotického systému

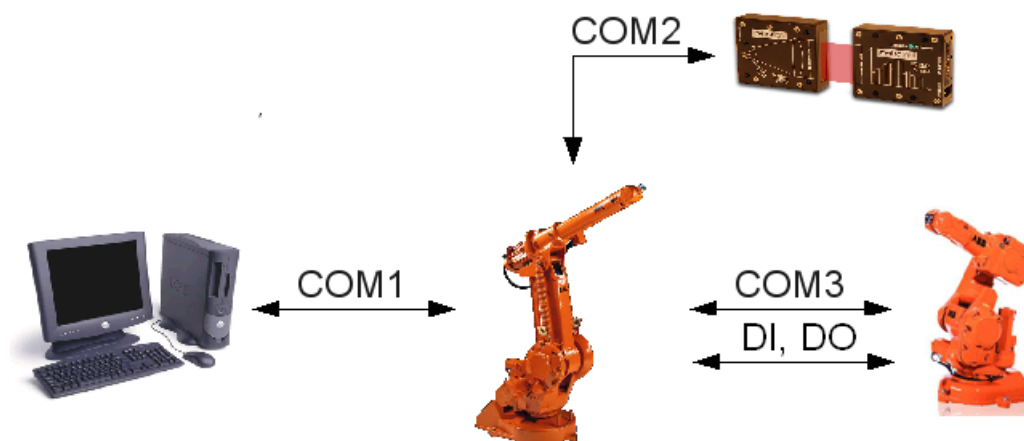
Tento úsek práce je vymezen pro popis komunikace mezi všemi zařízeními použitými pro realizaci úlohy. V jednotlivých částech jsou popsány přesné parametry nastavení řídicích systémů robotů pro správný chod celé úlohy.

Oba roboty mají shodný řídicí systém S4C+, který obsahuje dva porty RS232 a jeden RS422. První port RS232 označením „COM2“ je umístěn na komunikační desce s označením „DSQC 504“. Na stejné desce se nachází i port druhý port RS422 s označením „COM3“ (viz obrázek 13 a)). Další port RS232 se nachází v průmyslovém počítači uvnitř řídicího systému s označením „COM1“ (viz obrázek 13 b)). Oba roboty spolu ještě pro synchronizaci pohybů budou komunikovat pomocí digitálních vstupů a výstupů. Propojení celého systému je řešeno níže naznačeným způsobem.

**Znázornění propojení systému:**

- IRB 1400  $\longleftrightarrow$  COM2 (RS232)  $\longleftrightarrow$  snímač
- IRB 1400  $\longleftrightarrow$  COM3 (RS422)  $\longleftrightarrow$  IRB 140
- IRB 1400  $\longleftrightarrow$  COM1 (RS232)  $\longleftrightarrow$  PC

Schéma propojení komunikace jednotlivých zařízení (viz obrázek 12)



Obrázek 12: Schéma propojení celého systému

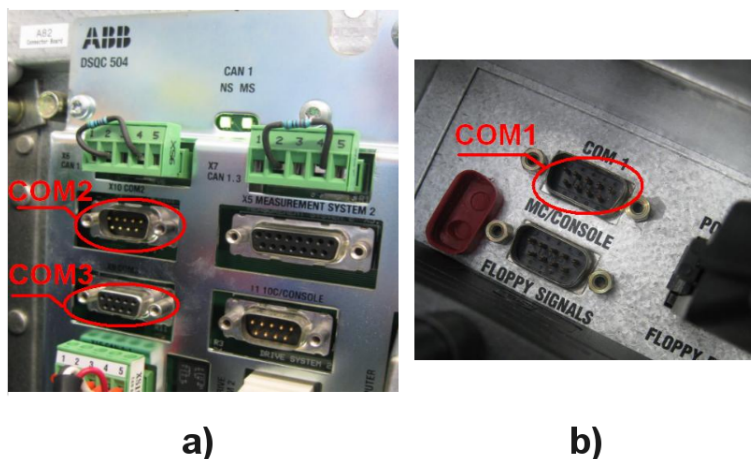
### 4.0.4 Komunikace mezi snímačem a robotem IRB 1400

Pro komunikaci mezi robotem IRB 1400 a snímačem je použito sériové rozhraní RS232. Pro snímač budeme využívat port s označením „COM2“ (viz obrázek 13 a)).

## Parametry komunikace řídicího systému

- Přenosová rychlost: 38400 b/s.
- Datové bity: 8
- Parita: žádný
- Řízení toku: žádný

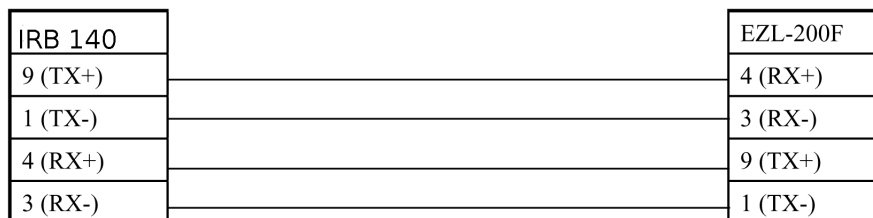
Postup pro nastavení sériového portu v řídicím systému robota (viz část 4.0.7).



Obrázek 13: Umístění portu COM2 a COM3 obrázek a), port COM1 obrázek b)

### 4.0.5 Komunikace mezi roboty IRB 140 a IRB 1400

Komunikace mezi oběma roboty bude probíhat pomocí sériového rozhraní RS422. U obou robotů je toto rozhraní označeno portem „COM3“. V řídicím systému obou robotů se nacházejí na shodném místě (viz obrázek 13 a)). Pro komunikaci musí být také samozřejmě připojen kabel se správným propojením pinů RS422 dle obrázku schéma zapojení 14.



Obrázek 14: Schéma propojení pinů dle RS422

### **Parametry nastavení komunikace mezi řídícími systémy (u obou shodně)**

- Přenosová rychlost: 38400 b/s.
- Datové bity: 8
- Parita: žádný
- Řízení toku: žádný

Postup pro nastavení sériového portu v řídícím systému robota (viz část 4.0.7).

#### **4.0.6 Komunikace mezi robotem IRB 1400 a PC**

Přenos dat mezi IRB 1400 a PC bude probíhat skrze rozhraní RS232 a port „COM1“ (viz obrázek 13 b)).

### **Parametry nastavení komunikace mezi řídícím systémem a PC**

- Přenosová rychlost: 19200 b/s.
- Datové bity: 8
- Parita: žádný
- Řízení toku: žádný

Postup pro nastavení sériového portu v řídícím systému robota (viz část 4.0.7).

#### **4.0.7 Nastavení sériového rozhraní v řídícím systému S4C+**

Tato část popisuje postup nastavení parametrů sériového rozhraní RS232 a RS422 v řídícím systému S4C+ robota IRB 140 a IRB 1400 pomocí ovládacího panelu, který se označuje názvem Teach Pendant (TP).

#### **Postup nastavení:**

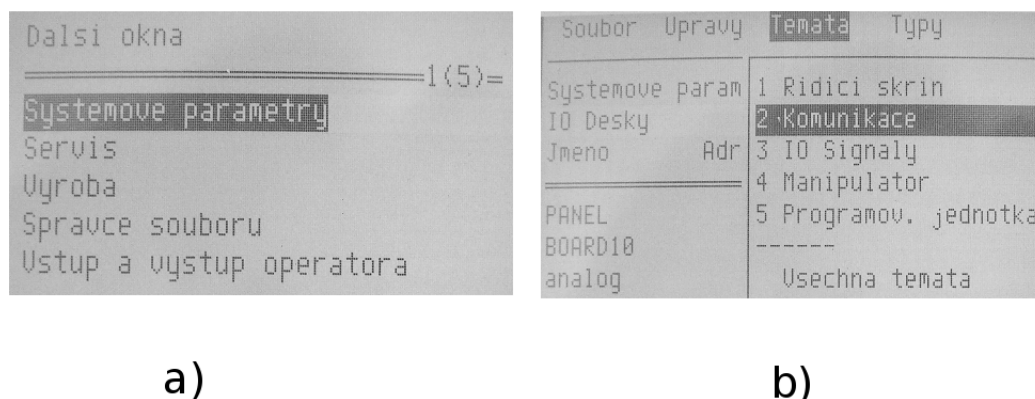
1. Na TP zvolíme tlačítko „Další okna“.



Obrázek 15: Ikona tlačítka „Další okna“

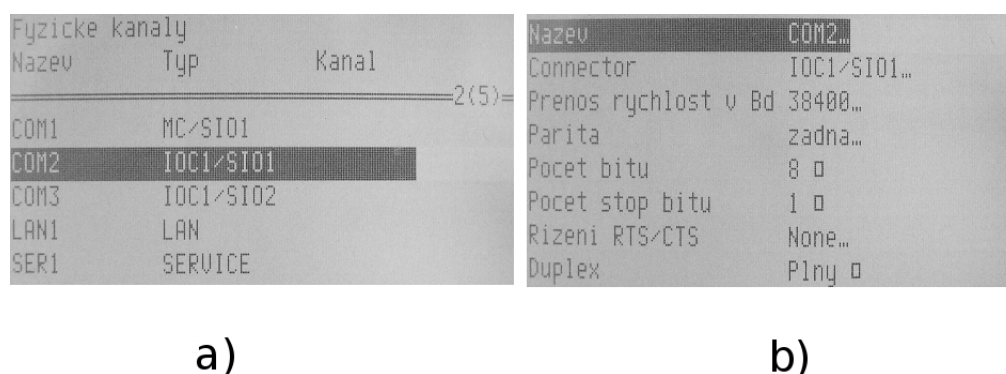


2. V seznamu vybereme položku „Systémové parametry“ (viz obrázek 16 a)).
3. V horní části zvolíme nabídku „Témata“ a druhou položku „Komunikace“ (viz obrázek 16 b)).



Obrázek 16: Teach Pendant - Položka systémové parametry obr. a), Položka komunikace obr. b)

4. Vybereme rozhraní s názvem „COM1“, „COM2“ nebo „COM3“ dle potřeby nastavení zařízení a potvrdíme entrem (viz obrázek 16 a)).
5. Nastavíme všechny parametry rozhraní (viz obrázek 16 b)).

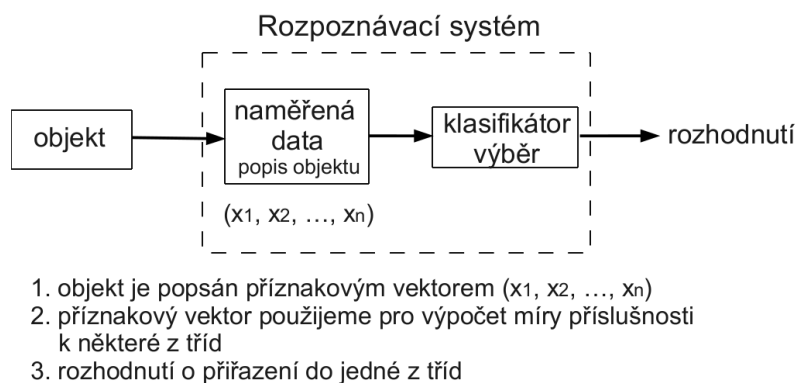


Obrázek 17: Teach Pendant - Položka výběr komunikačního rozhraní obr. a), Položka nastavení parametrů rozhraní obr. b)

## 5 Systém rozpoznávání předmětů

V této kapitole se dozvíme, jakým způsobem je prováděno rozpoznávání předmětů a jejich klasifikace pomocí snímače MetraLight. Na několika příkladech si i prakticky ukážeme funkčnost popsaného systému rozpoznávání. Dále si ukážeme způsob měření rozměrů a určování orientace u detekovaných předmětů. Celá tato kapitola je jednou ze stěžejních částí celé práce a proto je zapotřebí jí věnovat zvýšenou pozornost.

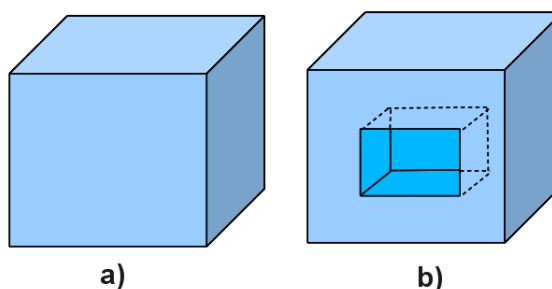
Všeobecný systém rozpoznávání dokáže rozhodnout a zařadit předmět mezi rozpoznané pouze tehdy, když ho podle nějakých příznaků dokáže identifikovat (viz obrázek 18). Systém je potřeba naučit jaké předměty má rozpoznávat a dle toho musí být schopen vyvodit závěr, zda předmět rozpozná nebo nikoliv. Každý předmět musí mít pro své rozpoznání definovány jednoznačné příznaky, pomocí kterých ho identifikuje.



Obrázek 18: Klasifikátor založený na příznakovém popisu

Dříve, než začneme systém navrhovat, si musíme položit otázku, jaké předměty jsme schopni s použitým zařízením detekovat. Omezení, která přináší použitá technologie měření spočívá v nemožnosti rozpoznávat jiné předměty než konvexního charakteru. Konvexním tělesem chápeme takové těleso, které má jen roviny, boule, hrany a rohy (viz obrázek 19 a)), ale nemá žádné kouty, důlky a ani dutiny (viz obrázek 19 b)) - konkávní těleso. Při výběru předmětu rozpoznávání jsme také omezeni velikostí konstrukce nesoucí snímač a v neposlední řadě také dosahem ramene robota.

Při popisu navržené metody budeme postupovat od základních tvarů rozpoznávaných těles, jako je kvádr nebo krychle, až po složitější a členitější tělesa s různou velikostí průřezu, čímž může být koule nebo kužel.



Obrázek 19: Konvexní těleso a), Konkávní těleso b)

## 5.1 Detekce předmětu pomocí přímek

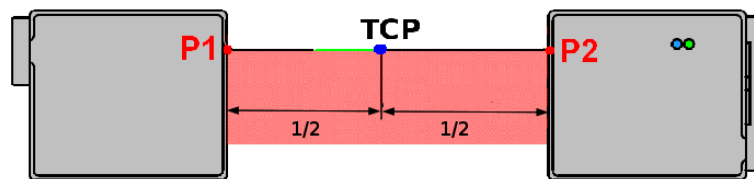
Způsob detekce objektu pomocí přímek je jednou z hlavních možností, jak daný snímač využít pro tuto úlohu. Pro zrealizování tohoto postupu je zapotřebí vhodně využít měřících funkcí použitého snímače a správně je zkombinovat s pohyby robota provádějícího detekci.

Z kapitoly věnované snímači MetraLight (viz 1.4) jsme si uvedli hlavní módy, které budeme využívat pro detekci objektů. Jedná se o hranové módy, kdy se měří vstupní (edge1, v programu mód m0) a výstupní (edge2, programu mód m1) hrana detekovaného objektu. Dále v části věnované jazyku Rapid jsme si uvedli, že je možné pomocí programového přerušení vykonávat více instrukcí najednou, čímž navodíme simulaci vícevláknového chodu. Při detekci by bylo vhodné, abychom se pohybovali ramenem robota a zároveň testovali přítomnost objektu v rozsahu snímače. Po detekci objektu bychom získali velikost přerušení paprsku vysílaného mezi oběma částmi snímače.

### 5.1.1 Definování bodů na snímači

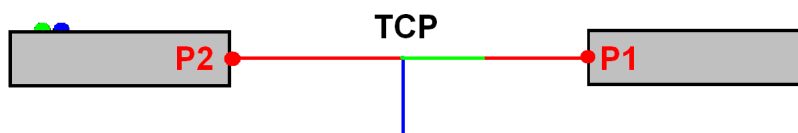
Nejprve je potřeba určit kde a kdy se budou přímky vytvářet. V eukleidovské geometrii existuje pro každé dva body právě jedna přímka, která oba prochází. Tato přímka představuje nejkratší spojnici mezi dotyčnými body a právě ta spojuje body na snímači. Definice bodů tvořící počátek a konec detekční přímky je velice důležitou částí celého systému rozpoznávání. Oba body tvořící přímku jsou přímo vztažené k TCP a provádějí stejné rotace a posuny jako celý nástroj. Přesnou pozici těchto bodů vypočítáme z natočení a polohy TCP pomocí transformačních matic popsaných v kapitole (viz 2.2.1). Pozici a aktuálního natočení TCP v prostoru získáme pomocí instrukce jazyka Rapid (viz 3.4.2). V programu jsou tyto body vytvořeny pomocí cílových bodů robtarget. Umístění TCP je uprostřed mezi oběma částmi

snímače na počátku měřicího rozsahu. Zobrazení počátečního umístění bodů  $P1$  a  $P2$  na snímači názorně (viz obrázek 20).



Obrázek 20: Umístění bodů: pohled shora

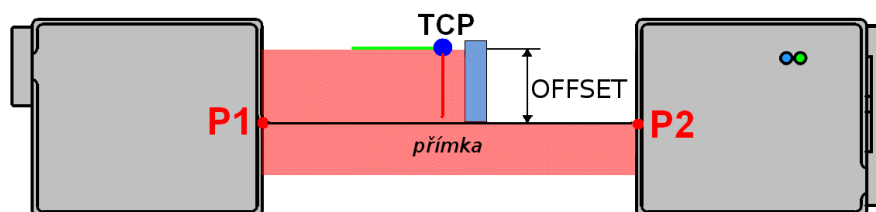
TCP a body by měly být definované ve stejné výšce a to ideálně ve výšce umístění laseru na snímači. V případě rozdílných výšek krajních bodů by později mohly vznikat přímky horizontálně nakloněné, které by měly za následek nepřesnosti při rozpoznání předmětu. Obrázek bočního pohledu na umístění bodů  $P1$ ,  $P2$  a TCP na snímači (viz 21).



Obrázek 21: Umístění bodů: boční pohled

### 5.1.2 Princip generování přímek

Pro generování přímek využijeme bodů, které jsme definovaly na snímači. Při vytváření se mohou body vygenerovat vždy po celém měřicím rozsahu snímače. Pozici vygenerované přímky určuje první naměřená vzdálenost (offset) předmětu vstupujícího a následně vystupujícího z rozsahu snímače (názorně viz obrázek 22). Velikost offsetu pak určí vzdálenost přímky (resp. bodů tvořící přímku) od TCP a určí se její pozice v prostoru.



Obrázek 22: Umístění bodů při detekci objektu: pohled shora (režim edge1)

### 5.1.3 Metoda generování přímek - trajektorie pohybu

Pro rozpoznávání předmětů je důležité se co nejvěrněji přiblížit tvaru skutečného předmětu. Je potřeba zvolit co nejuniverzálnější postup rozpoznávání, aby ho bylo možné aplikovat na co nejvíce tvarů předmětů a mohl mít co nejširší využití. Systém musí samozřejmě splňovat určité ergonomické předpoklady na časovou náročnost, kterou se snažíme minimalizovat pro co nejefektivnější chod celé aplikace.

Pro maximálně efektivní chod aplikace by bylo vhodné, kdyby se robot se snímačem pohyboval co nejrychleji a tím bychom byli schopni v minimálním možném čase rozpoznat tvar předmětu. Bohužel zde máme jedno významné omezení a to se nachází v řídicím systému S4C+, robota IRB 1400. Omezení spočívá v nemožnosti komunikovat se snímačem vícekrát, než 4-krát za sekundu pomocí programového přerušení. To znamená, že zde existuje horní hranice maximální rychlosti pohybu robota při detekci celé plochy, kde se bude nacházet hledaný objekt. V případě vyšší rychlosti, než maximální možné pro detekci by robot nestihl detekovat celý prostor pro hledání objektu a vznikaly by tak neprozkoumaná místa.

#### Výpočet maximální rychlosti robota při detekci vychází z:

1.  $s$  - šířky laserového paprsku snímače = 28 mm
2.  $p$  - počtu přístupů ke snímači za 1 sekundu = 4

#### Vzorec pro výpočet rychlosti pohybu robota při detekci:

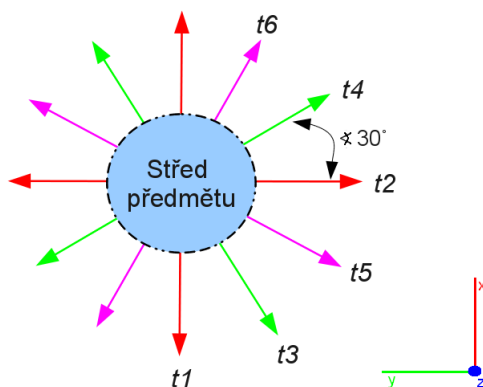
$$V_{robot} = s \cdot p = 112mm/s \quad (6)$$

Teoreticky maximální rychlost pohybu robota na řídicím systému S4C+ při detekci je 112 mm/s. Ovšem podporované rychlosti robota jsou mimo jiné (... , 80 mm/s, 100 mm/s, 150 mm/s, ...). Z výběru definovaných rychlostí jasně vyplývá, že nejvyšší rychlost při detekci můžeme použít 100 mm/s (v jazyku Rapidu značeno v100).

#### Trajektorie pohybu:

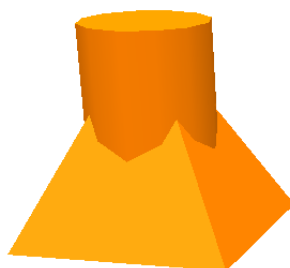
Při rozpoznávání předmětu použijeme metodu založenou na skenování 6-ti trajektorií, která se dále ukazuje jako postačující pro většinu zvolených úloh. Každý krok obsahuje projetí jedné trajektorie pod určitým úhlem natočení v ose  $z$ , vždy jiným. Při projíždění trajektorií jsou každé dvě na sebe kolmé. Pak vznikne jejich křížení uprostřed hledaného předmětu, kde jsou na sebe každé sousední trajektorie natočené po 30 stupních (viz obrázek 23). Během pohybu po všech trajektoriích se

budou generovat přímký vždy po vstupu a výstupu detekovaného předmětu z rozsahu laserového paprsku snímače. Podrobnější popis jednotlivých kroků detekce pomocí 6-ti trajektorií je popsán v části (viz 5.1.3). Tento postup byl navržen s ohledem na dostatečně velký počet detekovaných tvarů předmětů se snahou o minimální časovou náročnost.



Obrázek 23: Znázornění detekčních trajektorií

Při rozpoznávání předmětů budeme určovat tvar řezu prostředkem předmětu. Dále pro co největší počet rozpoznávaných předmětů budeme detekovat průměr předmětu v jeho spodní a horní části. Samozřejmě by šlo rozpoznávat i tvar předmětu v jeho spodní a horní části, ale to by pak byl celý systém rozpoznávání zbytečně časově náročnější. Příklad rozdílného tvaru průřezu předmětu ve spodní a horní části detekovaného předmětu (viz obrázek 24).

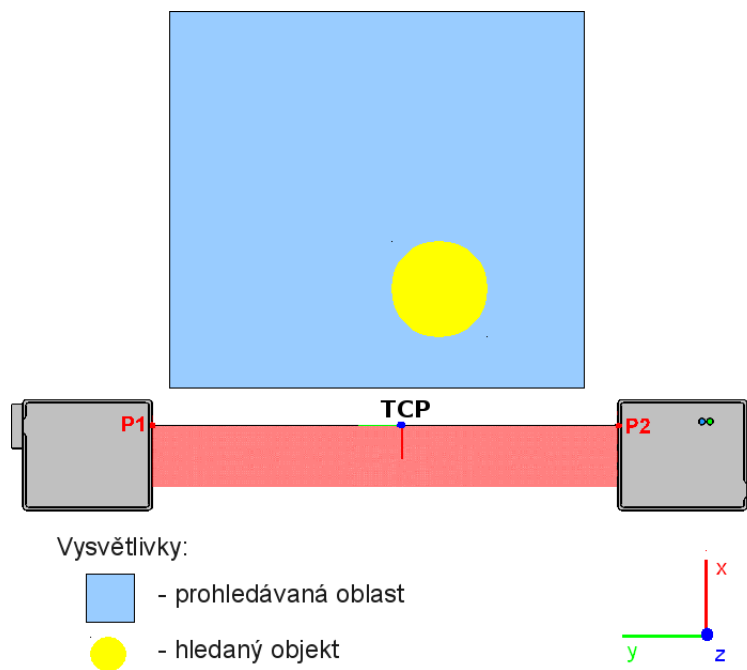


Obrázek 24: Příklad předmětu s rozdílnými tvary průřezů ve spodní a horní části

**Pro rozpoznávání základních tvarů tedy budeme předpokládat, že je tvar řezu předmětu ve všech částech stejný.** Na rozdíl od jeho průměru, který se může měnit. Příkladem může být koule nebo jehlan (viz obrázek 37). Dále si ještě jednou připomeneme, že detekovaná tělesa jsou konvexního tvaru (viz obrázek 19 a)). Pro celý tento postup je potřeba vhodným způsobem určit pořadí projíždění trajektorií robota provádějícího detekci.

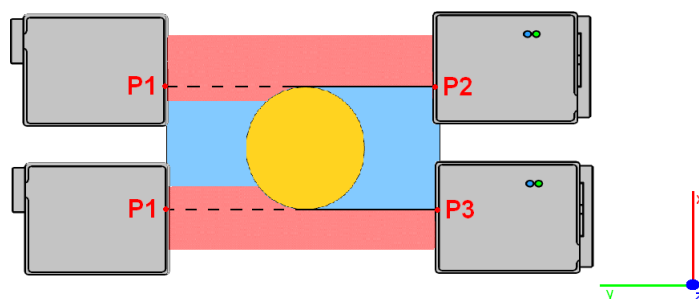
### Algoritmus řešení při prohledávání trajektorií:

Nyní si v jednotlivých bodech vysvětlíme, jak bude probíhat postupná detekce celého předmětu. Každý bod obsahuje důležitou část algoritmu, pomocí kterého získáme potřebná data pro následné rozpoznávání tvaru předmětu.



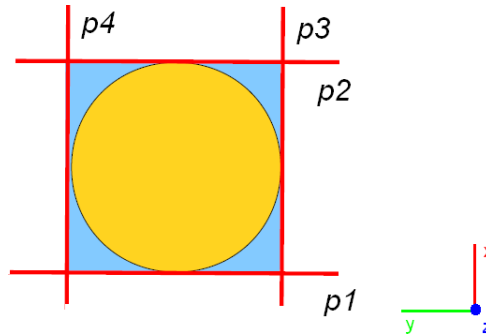
Obrázek 25: Počáteční stav systému

1. **Hledání objektu v ose x:** V prvních dvou bodech je nejprve potřeba najít objekt v hledaném prostoru a tím vymezit minimální plochu, kde by se mohl nacházet. To provedeme projetím trajektorie  $t1$  v přímém směru od robota v ose x s orientací nástroje  $(0,180,0)$  a prohledávání trajektorie. Tím omezíme hledaný prostor v ose x na minimální interval (viz obrázek 26).



Obrázek 26: Detekce předmětu v prostoru v ose x

2. **Hledání objektu v ose y:** Následujícím krokem je omezení prohledávaného prostoru na interval v ose y. To provedeme natočením nástroje na  $(180, 0, -90)$  a prohledáním trajektorie  $t2$  v ose y směrem od robota. Tím nám vznikne druhý interval pro osu y. Nyní máme omezený rozsah v osách x a y, kde se nachází hledaný objekt. Pomocí toho určíme přibližnou pozici středu rozpoznávaného předmětu, který se skrývá ve čtvercovém či obdélníkovém prostoru. Omezení intervalu v ose x a y (viz obrázek 27).



Obrázek 27: Ohraničení prostoru kde se nachází objekt. Generování přímek  $p1$ ,  $p2$ ,  $p3$ ,  $p4$

3. **Měření výšky předmětu v ose z:** Dalším krokem bude měření výšky předmětu, které provedeme detekcí pomocí příkazu (viz 2). Toto měření provádíme kvůli nalezení poloviny výšky předmětu, kde se bude provádět rozpoznávání tvaru řezu předmětu. Důvod měření ještě názorně na obrázku (viz 37). Výška předmětu se vypočítá jako rozdíl naměřené pozice koncového bodu TCP indikující vršek předmětu a výšku plochy, na které je měřený objekt položen (viz obrázek 28). Pozice TCP se počítá vzdáleností od aktuálního WObj.

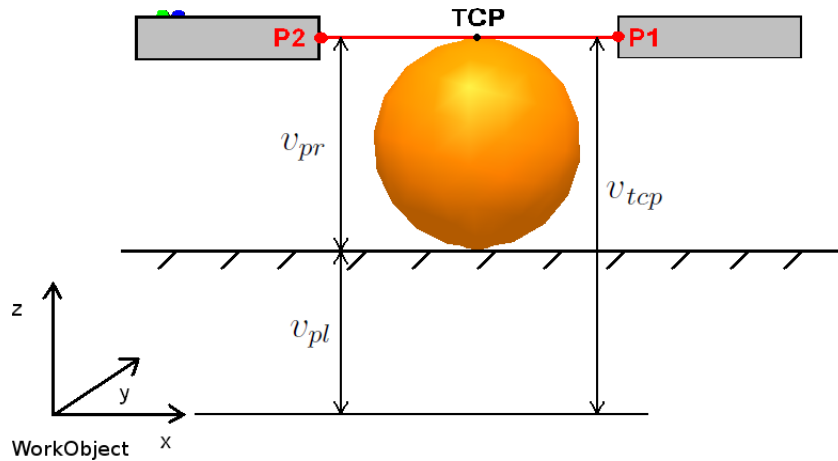
**Výšku předmětu vypočítáme jako:**

$v_{tcp}$  - pozice TCP u vršku předmětu vzhledem k WObj,

$v_{pl}$  - výška plochy,  $v_{pr}$  - výška předmětu

$$v_{pr} = v_{tcp} - v_{pl} \quad (7)$$

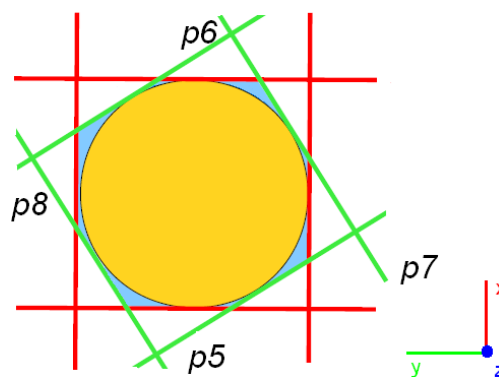




Obrázek 28: Měření výšky předmětu

4. Následující kroky už budou sloužit k přímému postupu rozpoznávání řezu předmětu pomocí skenování 6-ti trajektorií (viz obrázek 23)

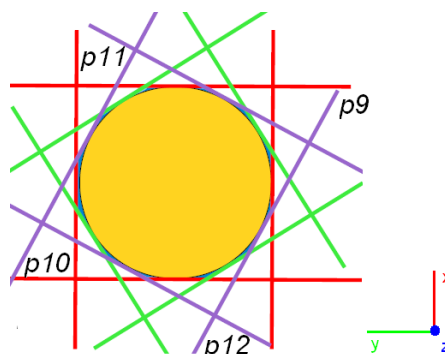
- Opět provedeme detekování v trajektoriích  $t1$  a  $t2$  (stejně jako v 1. a 2. bodě), ale s novou polohou snímače v ose  $z$  (polovina velikosti předmětu). Zde při každém novém výskytu hrany (vstupující nebo vystupující z rozsahu snímače) objektu vygenerujeme detekční přímkou  $p1$ ,  $p2$ ,  $p3$ ,  $p4$  (viz obrázek 27).
- Dalším postupem bude skenování trajektorie  $t3$  s orientací nástroje (180,0, -150) a vygenerování dalších dvou přímek  $p5$ ,  $p6$  (viz obrázek 29).
- Nyní další skenovací trajektorie  $t4$  kolmé k předchozí trajektorii, tedy natočení (-180, 0, 120) a generování dalších přímek  $p7$ ,  $p8$  (viz obrázek 29).



Obrázek 29: Generování přímek  $p5$ ,  $p6$ ,  $p7$ ,  $p8$  pro řez předmětu

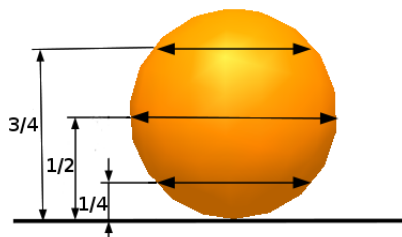
- Dalším krokem natočení nástroje je na (180,0,-120) a následné skenování trajektorie  $t5$  a generování přímek  $p9$ ,  $p10$  (viz obrázek 30).

- Následně kolmé k předchozí s orientací  $(-180,0,150)$  a prohledáním  $t6$  a získání dalších přímk potřebných pro detekování objektu  $p11$ ,  $p12$  (viz obrázek 30).



Obrázek 30: Generování přímk  $p9$ ,  $p10$ ,  $p11$ ,  $p12$  pro řez předmětu

- Poslední dvě skenování proběhnou shodně jako v bodech 1. a 2., ovšem s jinou výškou v ose  $z$ . Tato prohledávání určí velikost plochy předmětu u jeho **spodní a horní části**. Spodní část je měřena ve výšce  $1/4$  velikosti předmětu a horní část ve výšce  $3/4$  velikosti předmětu (viz obrázek 31). Důvod měření je kvůli zjištění členitosti předmětu viz část 5.4).



Obrázek 31: Vyznačení výšek při měření průměru předmětu

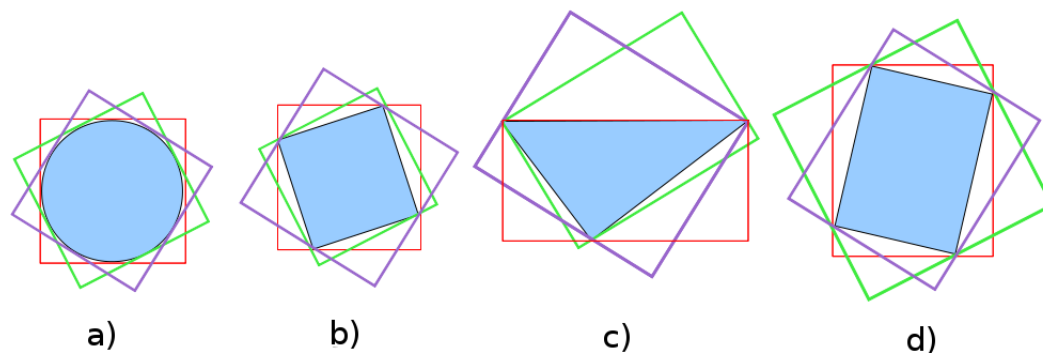
Nyní máme naměřená všechna potřebná data k rozpoznávání detekovaného objektu.

#### 5.1.4 Základní klasifikace tvarů předmětů

V této části budeme pokračovat ve zpracování naměřených dat, ze kterých vyvodíme základní rozdělení tvarů detekovaných objektů. Nejprve budeme zpracovávat přímky vygenerované v polovině výšky detekovaného objektu.

Pro základní klasifikaci předmětů můžeme nejprve využít tvary obrazců (dále detekční obrazce) vzniklých protínajícími se přímkami, vytvořenými v jednotlivých fázích detekce 6-ti trajektorií. Každý předmět má tři detekční obrazce: červený, zelený a fialový (viz obrázek 32).

Podle rozměrů stran jednotlivých detekčních obrazců můžeme nejjednodušším způsobem určit základní tvar řezu předmětu v jeho polovině. Příklady možných tvarů vzniklých detekčních obrazců: čtverec, obdélník nebo jejich kombinace (viz obrázek 32).



Obrázek 32: Tvary detekčních obrazců

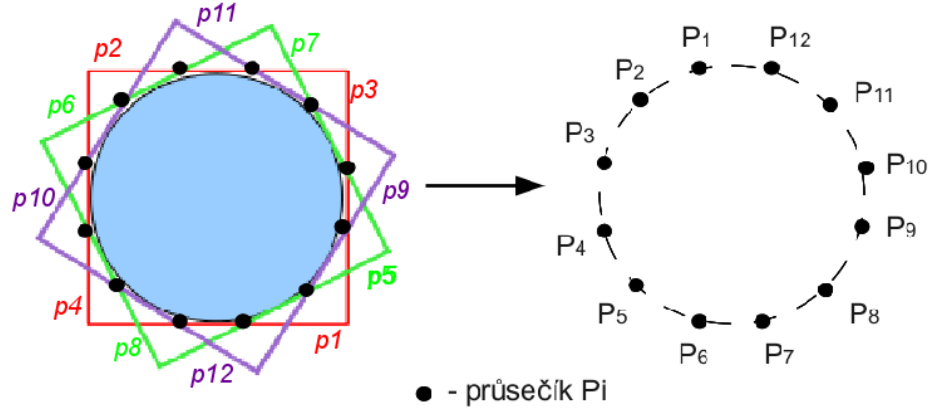
### **Základní kategorizace tvarů:**

1. Kategorie objektu 1 - všechny tři detekční obrazce mají tvar čtverce a shodný rozměr - kruhový tvar průřezu objektu (viz obrázek 32 a)). Typickým zástupcem tohoto tvaru může být válec.
2. Kategorie objektu 2 - všechny tři detekční obrazce mají tvar čtverce a různé rozměry - čtvercový tvar průřezu objektu (viz obrázek 32 b)) Typickým zástupcem může být například krychle.
3. Kategorie objektu 3 - alespoň jeden detekční obrazec je ve tvaru obdélníku ze tří obrazců - trojúhelníkový tvar nebo obdélníkový tvar průřezu objektu (viz obrázek 32 c) a d)) Typickým zástupcem může být trojboký hranol nebo kvádr.

Z předchozího rozdělení je vidět, že první dvě kategorie přesně definují tvar průřezu předmětu, ovšem třetí kategorie už nám jednoznačně neurčí tvar průřezu. Navíc tímto elementárním řešením neurčíme rozměry předmětu, a proto by bylo vhodné tuto metodu rozšířit. Jako další možnost se nabízí využít nějakým jiným, lepším způsobem vygenerované přímky.

## 5.2 Výpočet průsečíků přímek

Z vygenerovaných přímek by bylo vhodné nějakým univerzálním způsobem určit obrys a následně tvar průřezu předmětu. Z několika analýz bylo zjištěno, že je vhodné využít průsečíků (značeno  $P_{(x,y,z)}$  - uspořádaná trojice) vnitřních přímek tvořící obrazce, které nám můžou výrazně usnadnit práci rozhodování. Každou přímku protínají dvě sousední přímky a vzniká tím 12 průsečíků po obvodu hledaného objektu. Celá myšlenka návrhu vychází především z obrázku (viz 33).



Obrázek 33: Naznačení počítaných průsečíků z přímek

Všechny průsečíky přímek jsou přehledně vypsány v tabulce č. 6.

$P_1$ - p2 protíná p6	$P_7$ - p5 protíná p1
$P_2$ - p6 protíná p10	$P_8$ - p9 protíná p5
$P_3$ - p10 protíná p4	$P_9$ - p3 protíná p9
$P_4$ - p4 protíná p8	$P_{10}$ - p3 protíná p7
$P_5$ - p12 protíná p8	$P_{11}$ - p11 protíná p7
$P_6$ - p12 protíná p1	$P_{12}$ - p2 protíná p11

Tabulka 6: Přehledný výpis počítaných průsečíků

**Vztah k výpočtu průsečíků použijeme:**

Přímka  $a$ , body  $A=(x_1, y_1, z_1)$ ,  $B=(x_2, y_2, y_2)$

Přímka  $b$ , body  $C=(x_3, y_3, y_3)$ ,  $D=(x_4, y_4, y_4)$

$$p_x = \frac{(x_4 - x_3) \cdot (y_1 - y_3) - (y_4 - y_3) \cdot (x_1 - x_3)}{(y_4 - y_3) \cdot (x_2 - x_1) - (x_4 - x_3) \cdot (y_2 - y_1)}$$

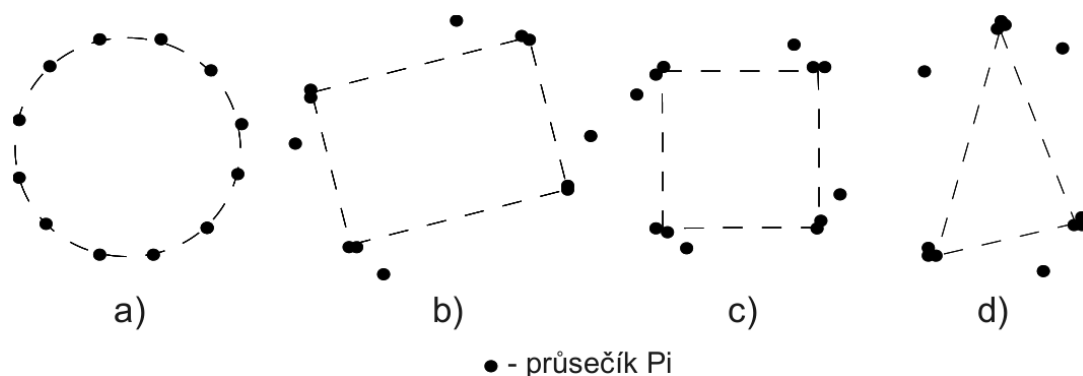
$$p_y = \frac{(x_2 - x_1) \cdot (y_1 - y_3) - (y_2 - y_1) \cdot (x_1 - x_3)}{(y_4 - y_3) \cdot (x_2 - x_1) - (x_4 - x_3) \cdot (y_2 - y_1)}$$

Průsečík existuje pokud platí:

$$p_y < 1, p_y > 0, p_x < 1, p_x > 0$$

$$\begin{aligned} P_{(x)} &= x_1 + p_x \cdot (x_2 - x_1) \\ P_{(y)} &= y_1 + p_y \cdot (y_2 - y_1) \\ P_{(z)} &= 1/2 \text{ výšky předmětu} \end{aligned} \quad (8)$$

Po výpočtu průsečíků pro detekovaný předmět vznikne obrys přibližného obrazce plochy obsazené objektem. Například pro objekt s kruhovým průřezem jsou průsečíky rovnoměrně rozmístěné po celém jeho obvodu (viz obrázek 34 a)). Naopak čtvercové, obdélníkové nebo trojúhelníkové průřezy obsahují po obvodu skupiny dvou, tří až čtyř průsečíků tvořící vrchol objektu (viz obrázek 34 b) c) d)).

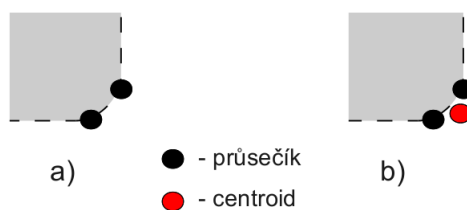


Obrázek 34: Rozmístění průsečíků při detekci například ve tvaru válce a), kvádru b), krychle c), trojboký hranol d)

Pomocí vypočítaných průsečíků dále můžeme určit přesné rozměry předmětu. Výpočet rozměrů jednotlivých předmětů (viz podkapitola 5.5).

### 5.3 Shluková analýza - metoda přijatelné vzdálenosti

Při aplikaci na reálném systému není vše tak ideální a průsečíky se přesně nepřekrývají. Příkladem může být předmět, který nemá ideálně zkosené hrany, ale mírně zaoblené. Průsečíky by se pak vyskytovaly v různých částech „zaobleného“ rohu předmětu, názorně (viz obrázek 35 a)). Proto je zapotřebí definovat metodu, která bude z blízkých bodů v přijatelné vzdálenosti vytvářet shluky. K přesnému určení rohu je zapotřebí ze získaných průsečíků tvořící shluk vypočítat jeden bod, který jim bude středem (dále nazýváno centroid, viz obrázek 35 b)).



Obrázek 35: Detail shluku tvořeného dvěma průsečíky a vypočítaného centroidu

**Shluk** - obsahuje alespoň dva průsečíky nacházející se od sebe v přijatelné vzdálenosti.

**Centroid** - je střední hodnota poloh průsečíků tvořící jeden shluk (značeno  $C_{(x,y,z)}$  - uspořádaná trojice)

**Přijatelná vzdálenost** - myslíme tím maximální dovolenou vzdálenost mezi dvěma body (průsečíky) tvořící jeden shluk. Vzdálenost je potřeba vhodně zvolit dle velikosti zkosení hran předmětů. Tato přijatelná vzdálenost dále určuje, jak přísný má být systém vytváření shluků a je zapotřebí ho vhodně zvolit. Maximální dovolenou vzdálenost vypočítáme pomocí Euklidovy vzdálenosti.

**Euklidova vzdálenost:** vypočítává vzdálenost, kterou mají mezi sebou dva body v prostoru nebo v rovině.

Vzorec pro výpočet: body  $A=(x_1,y_1,z_1)$ ,  $B=(x_2,y_2,z_2)$

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} \quad (9)$$

$d$  - je Euklidova vzdálenost mezi body  $A$  a  $B$ .

Metoda rozhodne, kolik shluků se v obrazu nachází a určí pro každý shluk svůj centroid.

### Jednotlivé kroky metody založené na přijatelné vzdálenosti

1. pro každý průsečík vypočítáme Euklidovy vzdálenosti ke všem zbylým průsečíkům
2. ke každému průsečíku vytvoříme tabulku, ve které budou vypsány průsečíky, které splňují podmínku přijatelné vzdálenosti
3. vytvoříme průniky tabulek z předchozího kroku a z nich shluky
4. pro každý shluk vypočítáme centroidy

### Postup pro výpočet centroidu:

1. počet průsečíků tvořící shluk 2 - centroid vypočítáme zprůměrováním obou pozic průsečíků (x,y,z) tvořící shluk (viz vzorec 10 a obrázek 36 a)).

$$n = 2, i \in N$$

$$P_{i(x)}, P_{i(y)}, P_{i(z)} \in P_{(x,y,z)}$$

$$C_{(x,y,z)} = \left( \frac{1}{n} \sum_{i=1}^n P_{i(x)}, \frac{1}{n} \sum_{i=1}^n P_{i(y)}, \frac{1}{n} \sum_{i=1}^n P_{i(z)} \right) \quad (10)$$

2. počet průsečíků tvořící shluk 3 - centroid vypočítáme zprůměrováním třech pozic (x,y,z) průsečíků tvořící shluk (viz vzorec 11 a obrázek 36 b)).

$$n = 3, i \in N$$

$$P_{i(x)}, P_{i(y)}, P_{i(z)} \in P_{(x,y,z)}$$

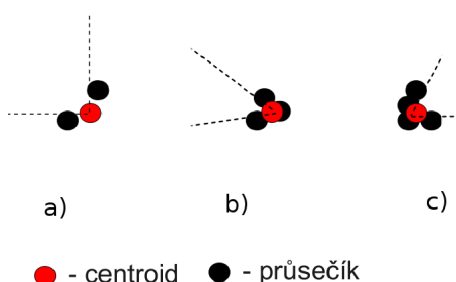
$$C_{(x,y,z)} = \left( \frac{1}{n} \sum_{i=1}^n P_{i(x)}, \frac{1}{n} \sum_{i=1}^n P_{i(y)}, \frac{1}{n} \sum_{i=1}^n P_{i(z)} \right) \quad (11)$$

3. počet průsečíků tvořící shluk 4 - centroid vypočítáme zprůměrováním čtyřech pozic (x,y,z) průsečíků tvořící shluk (viz vzorec (12 a obrázek 36 c)).

$$n = 4, i \in N$$

$$P_{i(x)}, P_{i(y)}, P_{i(z)} \in P_{(x,y,z)}$$

$$C_{(x,y,z)} = \left( \frac{1}{n} \sum_{i=1}^n P_{i(x)}, \frac{1}{n} \sum_{i=1}^n P_{i(y)}, \frac{1}{n} \sum_{i=1}^n P_{i(z)} \right) \quad (12)$$



Obrázek 36: Výpočet centroidu z průsečíků

Po výpočtu všech centroidů ze vzniklých shluků již můžeme přesně definovat všechny kategorie průřezu předmětu.

### Zpřesnění kategorizace tvaru průřezu:

1. Kategorie objektu 1 - všechny tři detekční obrazce mají tvar čtverce se shodným rozměrem a dále neobsahují žádný centroid - **kruhový tvar průřezu objektu**.
2. Kategorie objektu 2 - všechny tři detekční obrazce mají tvar čtverce s různými rozměry a má čtyři centroidy - **čtvercový tvar průřezu objektu**.
3. Kategorie objektu 3 - alespoň jeden detekční obrazec je ve tvaru obdélníku ze tří a má čtyři centroidy - **obdélníkový tvar průřezu objektu**.
4. Kategorie objektu 4 - alespoň jeden obrazec je ve tvaru obdélníku ze tří a má tři centroidy - **trojúhelníkový tvar průřezu objektu**.
5. Kategorie objektu 5 - je na tvarech detekčních obrazců nezávislá. Počet centroidů je méně než 3, nebo více než 4 - **neznámý tvar předmětu**.

Nyní jsme se dozvěděli, jaké kategorie předmětů daný systém rozeznává a jaké tvary průřezů předmětu dokážeme pomocí této metody určit.

## 5.4 Určení členitosti objektu

Pro další vylepšení předchozí metody a rozšíření množiny detekovaných předmětů v jednotlivých kategoriích využijeme provedená měření průměru ve spodní a horní části detekovaného předmětu (viz část 5.1.3). Z těchto měření detekčních obrazců určíme zda je předmět členitý či nikoliv.

**Členitost předmětu:** předmět je členitý, je-li jeho detekční obrazec měřený ve spodní ( $1/4$  výšky předmětu) nebo horní části ( $3/4$  výšky předmětu) rozměrově rozdílný vzhledem k detekčnímu obrazci změřeném v polovině jeho výšky.

Členitost určíme porovnáním velikosti spodního a horního detekčního obrazce vzhledem k detekčnímu obrazci prostředního řezu detekovaného předmětu (viz obrázek 37).

Pro značení členitosti předmětu je použito následující značení: (*spodní část, prostřední část, horní část*). Spodní a horní část může nabývat hodnot  $\{-1, 0, 1\}$  vůči jeho prostředku.

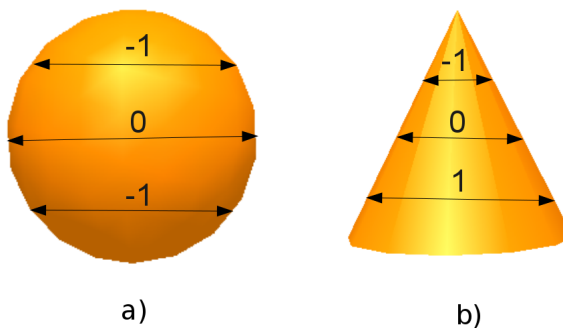
### Hodnoty členitosti objektu

- $-1$  - je menší



- 0 - je rovna
- 1 - je větší

Například koule -  $(-1,0,-1)$  nebo jehlan -  $(1,0,-1)$ , (viz obrázek 37).



Obrázek 37: Znárodnění značení členitosti předmětu koule a), kužel b)

#### 5.4.1 Rozpoznávané předměty - členitost

Po úspěšném detekování systém rozpoznává následující předměty ve čtyřech kategoriích včetně značení členitosti.

##### Kategorie objektu 1

- válec -  $(0,0,0)$
- koule -  $(-1,0,-1)$
- kužel -  $(1,0,-1)$

##### Kategorie objektu 2

- krychle -  $(0,0,0)$
- pravidelný čtyřboký jehlan -  $(1,0,-1)$
- hranol s podstavou čtverce -  $(0,0,0)$

##### Kategorie objektu 3

- nepravidelný čtyřboký jehlan -  $(1,0,-1)$
- hranol s podstavou obdélníka -  $(0,0,0)$
- válec položený na bok -  $(-1,0,-1)$

## Kategorie objektu 4

- trojboký hranol -  $(0,0,0)$
- trojboký jehlan -  $(1,0,-1)$

## 5.5 Výpočet rozměrů, orientace a pozice předmětu v prostoru

V této části si vysvětlíme, jakým způsobem bude probíhat výpočet všech důležitých rozměrů detekovaného předmětu a dále jeho orientace a pozice v prostoru z naměřených dat. V předchozí kapitole jsme provedli shlukovou analýzu, která nám poskytuje výstupní data do dalších výpočtů v této kapitole.

Pro výpočet rozměrů, orientace a pozice předmětu v prostoru budeme využívat naměřenou výšku detekovaného objektu a dále jeho centroidy tvořící řez prostředkem tělesa a detekční obrazce.

Orientaci předmětu budeme určovat podle natočení předmětu vzhledem k nulovému natočení robotu. Pozici předmětu budeme definovat pro osy x a y jako těžiště předmětu a v ose z jako vršek předmětu výpočtem z jeho výšky a plochy, na které se nachází. Při výpočtu se opět budeme řídit vytvořenými kategoriemi z předchozích kapitol, do kterých spadají rozpoznávané předměty. Při výpočtech se budeme snažit co nejvíce minimalizovat chybu měření průměrováním naměřených hodnot. U každého předmětu je ještě napsaný identifikační kód, který označuje každý tvar předmětu v programu.

### 5.5.1 Výpočet pro kategorii předmětů 1

Tělesa spadající do této kategorie: (válec, koule, kužel). Pro tuto kategorii se orientace předmětu nepočítá.

#### Popis výpočtu pro každé těleso:

- **Válec** - určující rozměry: *průměr podstavy* -  $d$ , *výška* -  $v$   
- průměr podstavy  $d$  - výpočet provedeme průměrem všech stran  $s_i$  detekčních obrazců tvořící řez v prostředku předmětu (viz obrázek 33).

$$\begin{aligned} n &= 12, \quad i \in N \\ d &= \frac{1}{n} \sum_{i=1}^n s_i \end{aligned} \tag{13}$$

- výška  $v$  - naměřená výška předmětu pomocí snímače (viz část 3).

Určení pozice předmětu (x,y,z) v prostoru:

Pozici středu v osách  $x$  a  $y$  získáme zprůměrováním všech průsečíků tvořící obvod předmětu (viz obrázek 33).

$$n = 12, i \in N$$

$$x = \frac{1}{n} \sum_{i=1}^n P_{i(x)}, \quad y = \frac{1}{n} \sum_{i=1}^n P_{i(y)} \quad (14)$$

Pozici v ose  $z$  vypočítáme jako: výška  $v$  + výška plochy, na které se předmět nachází, měřeno v souřadném systému (x,y,z) robota provádějícího rozpoznávání. Vše vychází z části (viz 3).

$v_{pr}$  - výška předmětu,  $v_{pl}$  - výška plochy

$$z = v_{pr} + v_{pl} \quad (15)$$

- kód předmětu: 0001

- **Koule** - určující rozměr: průměr -  $d$ .

- průměr  $d$  - výpočet proběhne shodným způsobem jako u válce, jen s tím rozdílem, že do průměrování započítáme ještě výšku předmětu.

$$n = 12, i \in N$$

$$d = \frac{1}{n+1} \left( \sum_{i=1}^n s_i + v \right) \quad (16)$$

Určení pozice předmětu (x,y,z) v prostoru:

Shodné jako u válce.

- kód předmětu: 0007

- **Kužel** - určující rozměry: průměr podstavy -  $d$ , výška -  $v$

- průměr podstavy  $d$  - vypočítá se stejným způsobem jako průměr válce a vynásobí se 2, jelikož se jedná o průměr uprostřed tělesa.

$$n = 12, i \in N$$

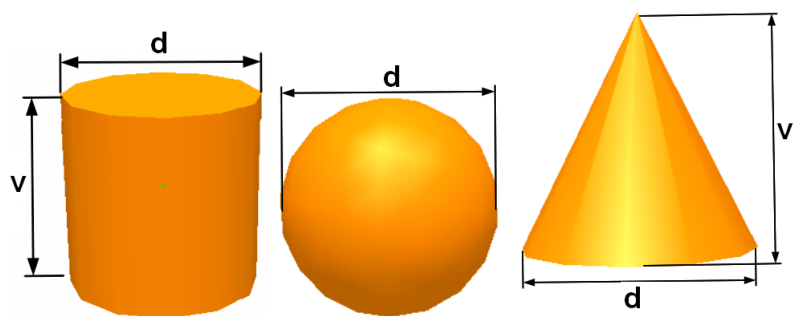
$$d = 2 \cdot \frac{1}{n} \sum_{i=1}^n s_i \quad (17)$$

- výška  $v$  - se vypočítá stejně jako v předchozích dvou případech.

Určení pozice předmětu (x,y,z) v prostoru:

Shodné jako u válce

- kód předmětu: 0006



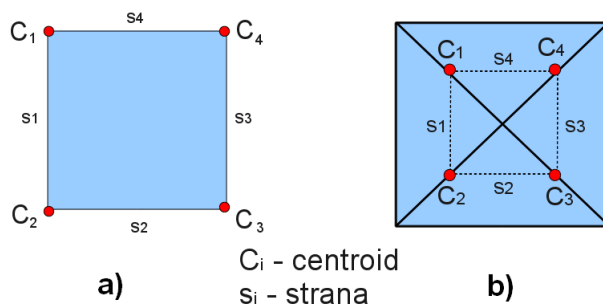
Obrázek 38: Obrázky předmětů z kategorie 1

### 5.5.2 Výpočet pro kategorii předmětů 2

Tělesa spadající do této kategorie: (krychle, pravidelný čtyřboký jehlan, hranol s podstavou čtverce). Pro tuto kategorii přibude určení orientace předmětu v prostoru.

**Popis výpočtu pro každé těleso:**

- **Krychle** - určující rozměry: *strana* -  $a$ , *orientace v prostoru*  $(x, y, z)$ .  
- *strana  $a$*  - vypočítáme ji jako průměr velikosti stran  $s_i$  obvodu průřezu předmětu a jeho výšky (viz obrázek 39 a)).



Obrázek 39: Označení stran krychle a), označení stran pravidelného jehlanu b)

$$n = 4, s_i \in N$$

$$a = \frac{1}{n+1} \left( \sum_{i=1}^n s_i + v \right) \quad (18)$$

Určení pozice předmětu  $(x, y, z)$  v prostoru: pro osy  $x$  a  $y$  proběhne vypočtením těžiště čtyřech centroidů po obvodu předmětu.

$$n = 4, i \in N$$

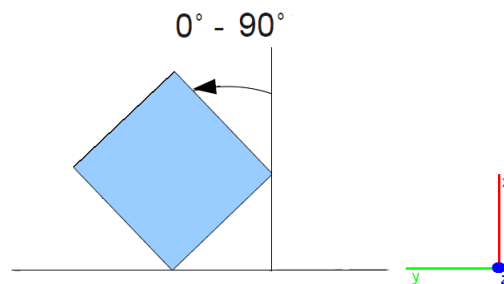
$$x = \frac{1}{n} \sum_{i=1}^n P_{i(x)}, \quad y = \frac{1}{n} \sum_{i=1}^n P_{i(y)} \quad (19)$$

Pozici v ose  $z$  vypočítáme jako: *strana  $a$  (výška předmětu)* + výška plochy, na které se předmět nachází, měřeno v souřadném systému  $(x,y,z)$  robota provádějícího rozpoznávání. Vše vychází z části (viz 3).

$v_{pr}$  - výška předmětu,  $v_{pl}$  - výška plochy

$$z = v_{pr} + v_{pl} \quad (20)$$

- orientace předmětu  $(x,y,z)$  - předmět se nachází v rovině a není nakloněný, proto budeme počítat pouze jeho natočení v ose  $z$ . Zbylé dvě osy  $x$  a  $y$  zůstávají nulové. Když se podíváme na krychli pohledem shora v ose  $z$ , představíme si ji jako 2D čtverec, kde jeho vrcholy tvoří centroidy (viz obrázek 40). Možnost natočení je v rozmezí  $0^\circ - 90^\circ$ , poté se rotace začnou opakovat. Vše je názorně na obrázku vyznačeno, který úhel se počítá.



Obrázek 40: Rotace předmětu s podstavou čtverce

- kód předmětu: 0003

- **Pravidelný čtyřboký jehlan** - určující rozměry: *velikost podstavy -  $d$ , výška -  $v$ , orientace v prostoru  $(x,y,z)$* .
  - velikost podstavy  $d$  - vypočítáme ji jako průměr velikosti stran  $s_i$  obvodu průřezu předmětu násobeno 2, jelikož se jedná o průměr uprostřed tělesa (viz obrázek 39 b)).

$$n = 4, s_i \in N$$

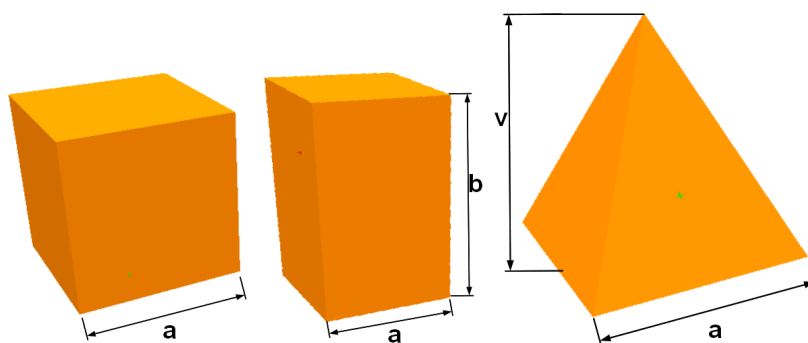
$$d = \frac{1}{n} \sum_{i=1}^n s_i \cdot 2 \quad (21)$$

- výška  $v$  - velikost získáme naměřením výšky předmětu pomocí snímače (viz kapitola 3).

Určení pozice předmětu  $(x,y,z)$  v prostoru: shodné jako u krychle

- Orientace předmětu  $(x,y,z)$  - vypočítá se jako v předešlém případě u krychle
- kód předmětu: 0005

- **Hranol s podstavou čtverce** - určující rozměry: *strana a*, *strana b*, *orientace v prostoru* ( $x, y, z$ ).
    - *strana a* - její rozměr vypočítáme podobně jako u krychle ovšem nezapočítáme výšku předmětu. Vypočítáme ji jako průměr vzdáleností stran  $s_i$  tvořící obvod průřezu předmětu (viz obrázek 39 a)).
    - *strana b* - bude vypočítána jako výška předmětu jako u předchozího tělesa
- Určení pozice předmětu ( $x, y, z$ ) v prostoru: shodné jako u krychle
- *orientace předmětu* ( $x, y, z$ ) - vypočítá se jako v předešlém případě u krychle
  - *kód předmětu*: 0022



Obrázek 41: Kategorie předmětů 2

### 5.5.3 Výpočet pro kategorii předmětů 3

Tělesa spadající do této kategorie: (hranol s obdélníkovou podstavou, nepravidelný čtyřboký jehlan, válec položený na bok). U této kategorie bude určena orientace předmětu v prostoru. Na příkladě si ukážeme, jak ji budeme určovat.

- **Hranol obdélníkovou podstavou** - určující rozměry: *strana a*, *strana b*, *strana c*, *orientace v prostoru* ( $x, y, z$ ).
  - *strana a* - vypočítáme ji jako průměr lichých stran  $s_i$  tvořící obvod objektu (viz obrázek 42 a)).

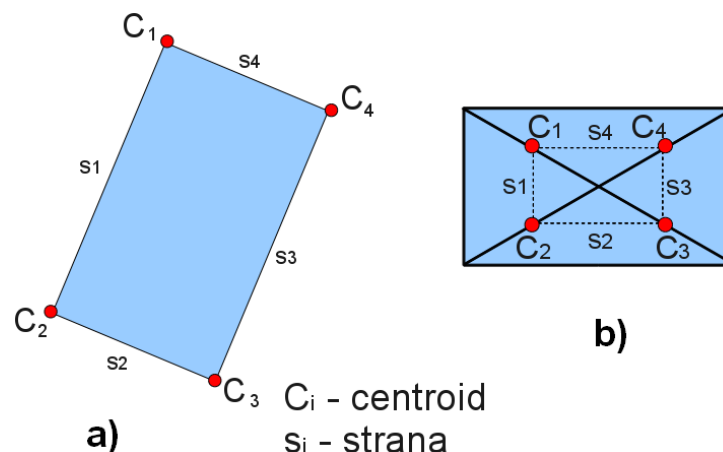
$$s_1, s_3 \in N$$

$$a = \frac{1}{2}(s_1 + s_3) \quad (22)$$

- *strana b* - vypočítáme ji jako průměr sudých stran  $s_i$  tvořící obvod objektu vynásobeno 2, jelikož to je průměr uprostřed tělesa (viz obrázek 42 a)).

$$s_2, s_4 \in N$$

$$b = \frac{1}{2}(s_2 + s_4) \quad (23)$$



Obrázek 42: Označení stran kváдру a), označení stran nepravidelného jehlanu b)

- strana c - vypočítá se jako naměřená výška předmětu pomocí snímáče (viz kapitola 3).

Určení pozice předmětu (x,y,z) v prostoru:

Pozici středu v osách x a y získáme zprůměrováním všech centroidů tvořící obvod (viz obrázek 42 b)).

$$n = 4, i \in N$$

$$x = \frac{1}{n} \sum_{i=1}^n C_{i(x)}, \quad y = \frac{1}{n} \sum_{i=1}^n C_{i(y)} \quad (24)$$

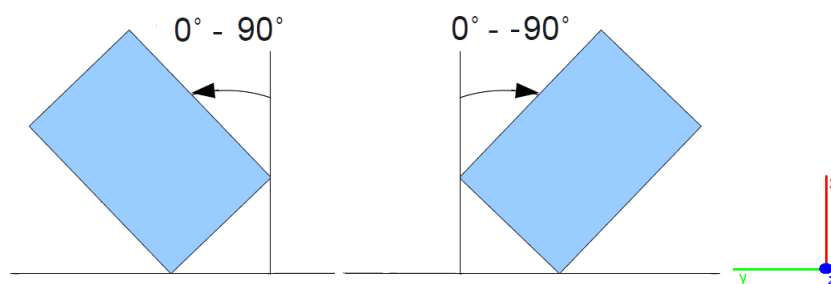
Pozici v ose z vypočítáme jako: *strana c (výška předmětu)* + pozice v ose z plochy, na které se předmět nachází, měřeno v souřadném systému (x,y,z) robota provádějícího rozpoznávání. Vše vychází z části (viz 3).

*v<sub>pr</sub> - výška předmětu, v<sub>pl</sub> - výška plochy*

$$z = v_{pr} + v_{pl} \quad (25)$$

- orientace předmětu (x,y,z) - předmět se nachází v rovině a, proto budeme počítat pouze jeho natočení v ose z. Zbylé dvě osy x a y zůstávají nulové. Když se podíváme na měřený hranol při pohledu shora v ose z, představíme si ho jako 2D obdélník, kde jeho vrcholy tvoří její centroidy (viz obrázek 43). Možnost natočení je v rozmezí 0° - 90° a 0° - -90°, poté se rotace začnou opakovat. Vše je názorně vyznačeno na obrázku, které úhly se počítají.

- kód předmětu: 0002



Obrázek 43: Rotace předmětu s podstavou obdélníka

- **Nepravidelný čtyřboký jehlan** - určující rozměry: *podstava a*, *podstava b*, *výška*, *orientace v prostoru* ( $x, y, z$ ).
  - podstava a - vypočítáme ji jako průměr velikosti lichých stran  $s_i$  tvořící obvod objektu vynásobeno 2, jelikož se jedná o průměr uprostřed tělesa (viz obrázek 42 b)).

$$s_2, s_4 \in N$$

$$a = \frac{1}{2}(s_1 + s_3) \cdot 2 \quad (26)$$

- podstava b - vypočítáme ji jako průměr velikosti sudých stran  $s_i$  tvořící obvod objektu vynásobeno 2, jelikož se jedná o průměr uprostřed tělesa (viz obrázek 42 b)).

$$s_1, s_3 \in N$$

$$b = \frac{1}{2}(s_2 + s_4) \cdot 2 \quad (27)$$

- výška v - vypočítá se jako naměřená výška předmětu pomocí snímáče (viz kapitola 3).

Určení pozice předmětu ( $x, y, z$ ) v prostoru:

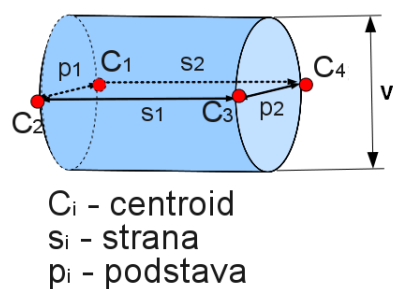
Shodné jako u hranolu s obdélníkovou podstavou.

- orientace předmětu ( $x, y, z$ ) - orientace se určí stejným způsobem jako v případě hranolu s obdélníkovou podstavou.

- kód předmětu: 0008

- **Válec položený na bok** - určující rozměry: *průměr podstavy* -  $d$ , *výška* -  $v$ , *orientace v prostoru* ( $x, y, z$ ).
  - průměr podstavy d - průměr podstavy určíme zprůměrováním *výšky v* předmětu a strany předmětu tvořící podstavu předmětu  $p_i$ , názorně (viz obrázek 44).





Obrázek 44: Válec položený na bok, znázornění stran pro výpočet

$$n = 2, s_i \in N$$

$$d = \frac{1}{n+1} \left( \sum_{i=1}^n p_i + v \right) \quad (28)$$

- výška  $v$  - určíme ji zprůměrováním vzdáleností stran  $s_i$ , které tvoří boční stěny válce, názorně (viz obrázek 44).

$$n = 2, s_i \in N$$

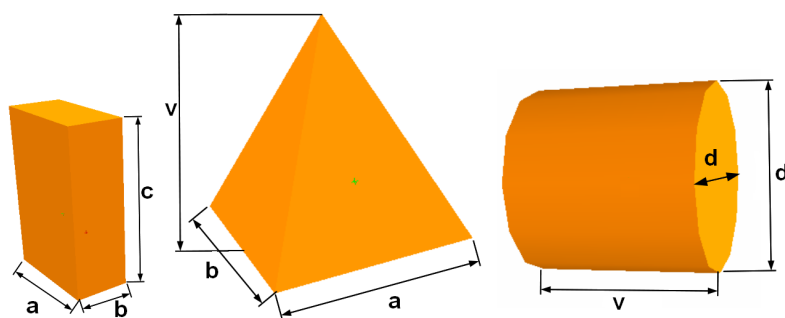
$$v = \frac{1}{n} \sum_{i=1}^n s_i \quad (29)$$

Určení pozice předmětu  $(x, y, z)$  v prostoru:

Shodné jako u hranolu s obdélníkovou podstavou.

- orientace předmětu  $(x, y, z)$  - orientace se určí stejným způsobem jako v případě hranolu s obdélníkovou podstavou.

- kód předmětu: 0011

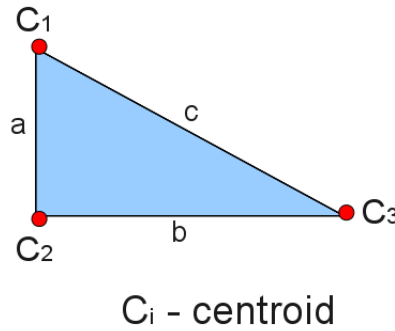


Obrázek 45: Tvary předmětů kategorie 3

#### 5.5.4 Výpočet pro kategorii předmětů 4

Tělesa spadající do této kategorie: (trojboký hranol, trojboký jehlan). Pro tuto kategorii budeme určovat orientaci předmětu.

- **Trojboký hranol** - určující rozměry: *strana - a*, *strana - b*, *strana - c*, *výška - v*, *orientace v prostoru (x,y,z)*.
  - strana a, strana b, strana c - jejich rozměry určíme postupným výpočtem vzdáleností centroidů tvořící jeho obvod dle vzorce pro Euklidovu vzdálenost 5.3, strany (viz obrázek 46).
  - výška v - vypočítá se jako naměřená výška předmětu pomocí snímáče (viz



Obrázek 46: Nepravidelný trojboký hranol, strany

kapitola 3).

Určení pozice předmětu (x,y,z) v prostoru:

Pozici středu v osách x a y získáme zprůměrováním všech centroidů tvořící obvod.

$$n = 3, i \in N$$

$$x = \frac{1}{n} \sum_{i=1}^n C_{i(x)}, \quad y = \frac{1}{n} \sum_{i=1}^n C_{i(y)} \quad (30)$$

Pozici v ose z vypočítáme jako: *výška v (výška předmětu)* + výška plochy, na které se předmět nachází, měreno v souřadném systému (x,y,z) robota provádějícího rozpoznávání. Vše vychází z části (viz 3).

$v_{pr}$  - *výška předmětu*,  $v_{pl}$  - *výška plochy*

$$z = v_{pr} + v_{pl} \quad (31)$$

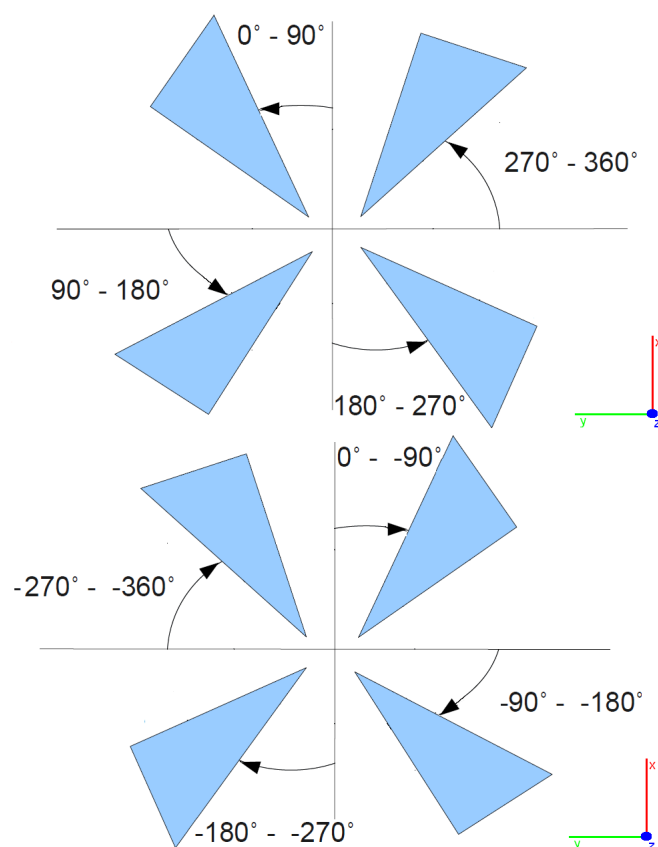
Určení pozice předmětu (x,y,z) v prostoru:

Shodné jako u trojbokého hranolu.

- orientace předmětu (x,y,z) - u trojbokého hranolu budeme předpokládat podstavu ve tvaru nerovnostranného trojúhelníku. Předmět se nachází v rovině, a

proto budeme počítat pouze jeho natočení v ose z. Zbylé dvě osy x a y zůstávají nulové. Když se podíváme na trojboký hranol při pohledu shora v ose z, představíme si ho jako 2D trojúhelník, kde vrcholy tvoří centroidy (viz obrázek 47). Možnost natočení je v rozmezí  $0^\circ - 360^\circ$  a  $0^\circ - -360^\circ$ . Natočení v ose z určujeme podle nejdelší strany. Vše je názorně na obrázku vyznačeno, které úhly se počítají (viz 47).

- kód předmětu: 0004



Obrázek 47: Rotace předmětu s podstavou trojúhelníka

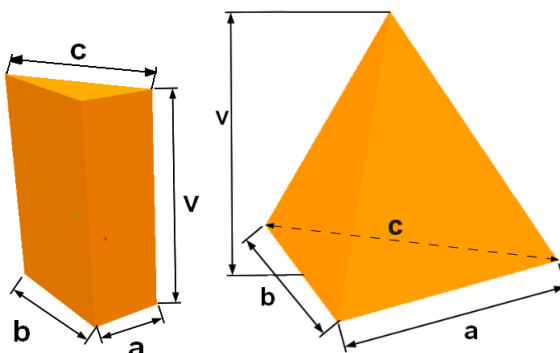
- **Trojboký jehlan** - určující rozměry: *podstava - a*, *podstava - b*, *podstava - c*, *výška - v*, *orientace v prostoru (x,y,z)*.  
- podstava a, podstava b, podstava c - se vypočítá postupným výpočtem vzdáleností centroidů tvořící jeho obvod stejně jako v předchozím případě, ovšem přibude vynásobení stran dvěma, jelikož to je průměr uprostřed tělesa.

$$\text{strana } a = \text{strana } a \cdot 2$$

$$\text{strana } b = \text{strana } b \cdot 2$$

$$\text{strana } c = \text{strana } c \cdot 2 \quad (32)$$

- výška  $v$  - výška se vypočítá jako u trojbokého hranolu.
- orientace předmětu  $(x,y,z)$  - se provede shodným způsobem jako u trojbokého hranolu.
- kód předmětu: 0044

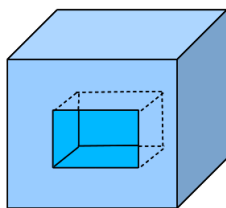


Obrázek 48: Kategorie předmětů 4

#### 5.5.5 Předměty nerozpoznávané systémem - kategorie předmětů 5

Předměty spadající do kategorie 5 jsou tvary, které jsou pro systém neznámé (nerozpoznatelné). Důvody vycházejí z technických předpokladů použitého zařízení. Dále z definice použité metody, která se zaměřuje na základní tvary předmětů. Mezi předměty nerozpoznatelné navrženým systémem mimo jiné patří:

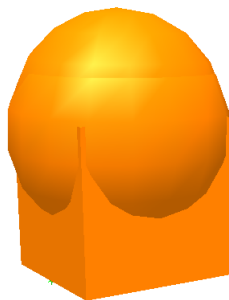
- všechna konkávní tělesa - tělesa obsahující: kouty, důlky a dutiny (viz obrázek 49)



Obrázek 49: Příklad konkávního tělesa s dutinou

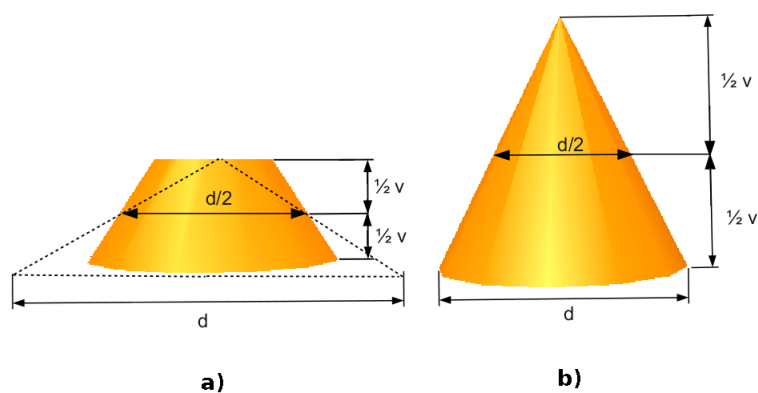
- hranoly pěti a víceboké - tento tvar by bylo možné rozpoznávat při rozšíření popsané metody o více vygenerovaných detekčních přímk po obvodu předmětu. Toto vylepšení, by ale přineslo větší časovou náročnost pro rozpoznávání předmětů, a proto nebylo implementováno.

- předměty různých tvarů průřezů v spodní a horní části (viz obrázek 50)



Obrázek 50: Různý tvar průřezu ve spodní a horní části

- nerozpoznává dále například komolá tělesa - při výpočtu velikosti podstavy vycházíme z předpokladu, těleso není komolé. Na obrázku (viz 51 a) a b)) je vidět, jak by postupoval algoritmus výpočtu, ve kterém by docházelo ke špatnému výpočtu velikosti podstavy  $d$  pro komolé předměty. Tento problém by bylo možné řešit pomocí poměrů detekčních obrazců ve spodní, prostřední a horní části předmětu.



Obrázek 51: Naznačení výpočtu podstavy například u komolého a) a nekomolého kuželu b)

- předměty o velikosti větší, než je rozpětí ramen konstrukce nesoucí snímač.

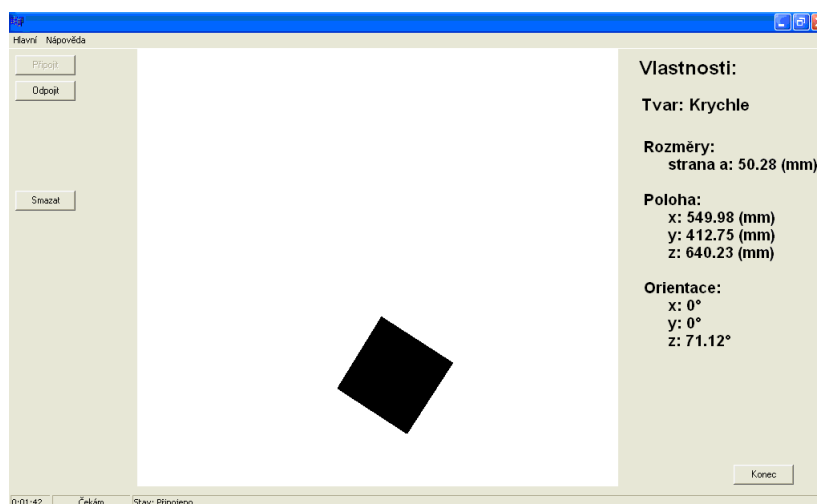
## 6 Vizualizační aplikace

V této části se dozvíme, jaké vlastnosti a funkce obsahuje vytvořený počítačový software pro zobrazování naměřených dat robotickým systémem.

Pro celý navržený robotický systém by bylo vhodné, kdyby se daly nějakým způsobem zobrazovat rozměry a tvary předmětů detekované snímačem. Vstup dat do této aplikace zajišťuje robot IRB 1400, který provádí rozpoznávání a vyhodnocování dat o předmětu. Robot komunikuje s PC pomocí sériového rozhraní RS232. Natavení komunikace je popsáno v části (viz 4.0.6). Výstupem vytvořené aplikace je výpis pozice, rozměrů a orientace předmětu v prostoru a dále vykreslení dvourozměrného tvaru detekovaného předmětu na plátně (pohled shora v ose z). Vytvořená aplikace je informativního charakteru, která má podávat obsluze stručný přehled o aktuálním detekovaném předmětu (viz obrázek 52). Nápad vytvořit vizualizační software vznikl po konzultaci s vedoucím práce.

### Popis funkcí ovládacích tlačítek:

- Připojit - připojí aplikaci k sériovému portu a čeká na příchozí data
- Odpojit - ukončení spojení aplikace k sériovém portu
- Konec - bezpečné ukončení aplikace
- Smazat - vymaže všechny údaje o naměřeném objektu



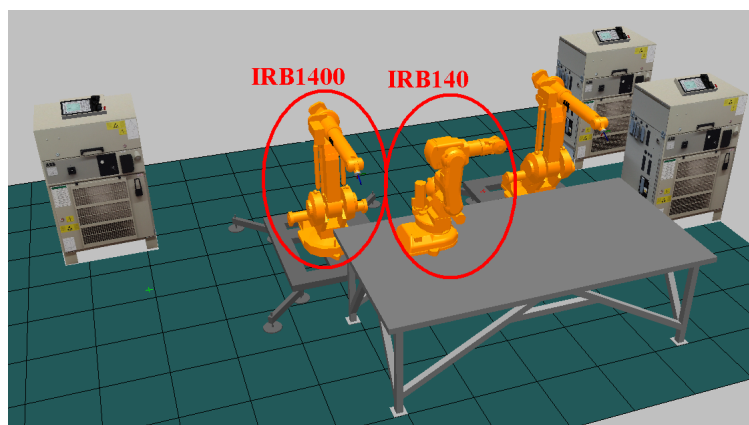
Obrázek 52: Náhled vytvořeného programu pro vykreslení a výpis naměřených dat

Celá aplikace byla vytvořena v programovacím jazyku C++ ve vývojovém prostředí Borland C++ Builder. Vytvořená aplikace součástí přiloženého CD.

## 7 Praktické řešení robotické aplikace

V této části se budeme zabývat vlastním praktickým řešením postupných činností systému rozpoznávání předmětů při vzájemné koordinaci pohybů obou robotů.

Je zde popsáno podrobně celé řešení úlohy, které bylo navrženo po získání znalostí z předchozích kapitol. Oba roboty používají pouze lineárního pohybu po trajektorii. Celá úloha byla prakticky odzkoušena v laboratoři inteligentních robotů na reálném robotickém systému. Vyznačení použitých robotů k práci v laboratoři inteligentních robotů je vidět na obrázku (viz 53) vymodelovaném v RobotStudios 4.0.



Obrázek 53: Model laboratoře inteligentních robotů

### 7.1 Popis jednotlivých činností navrženého systému:

Nyní si popíšeme celou funkci aplikace v jednotlivých krocích, jak probíhají na reálném systému. Ještě jednou si pro přehlednost připomeneme popis funkcí obou robotů.

Robot IRB 1400 provádí detekci, rozpoznávání a měření předmětu. Jeho nástroj je laserový snímač MetraLight (viz kapitola 1.2.1).

Robot IRB 140 provádí manipulační činnost. Robot bude provádět uchopení předmětu a přenesení na předem definované místo. Jeho nástroj je pneumatická přísavka (viz kapitola 1.1.1).

#### Činnost systému v postupných krocích:

1. Startem systému je synchronizace celého pracoviště pomocí digitálních vstupů a výstupů. Poté se oba roboty přesunou do výchozí domovské pozice.

2. Jako první začne provádět pohyb robot IRB 1400. Druhý robot IRB 140 čeká na synchronizační impuls. Robot nesoucí snímač nejprve začne provádět detekci předmětu v oblasti, kde by se měl nacházet předmět zájmu celého systému. Při detekci začne provádět jednotlivé kroky:

- detekce předmětu
- generování přímek
- rozpoznávání předmětu a jeho klasifikace
- výpočet rozměrů, polohy a orientace předmětu

Celý proces rozpoznávání je popsán (viz kapitola 5).

3. **Výsledek rozpoznávání může vydat dva výsledky:**

- a) Systém rozpozná předmět a zařadí do správné kategorie - program bude pokračovat dalším bodem 4)
- b) Systém nerozpozná předmět a nezařadí jej do žádné kategorie - Robot IRB 1400 se vrací do domovské pozice a čeká na zásah obsluhy, která by měla nedefinovaný předmět odstranit. Poté robot začne provádět všechny kroky systému znovu od začátku.

4. Po detekci a klasifikaci předmětu se řídicí systém robota IRB 1400 spojí pomocí rozhraní RS232 s PC určeným pro zobrazení naměřených a vypočítaných hodnot. Na něm je spuštěný software, vytvořený pro zobrazení příchozích dat. Data odeslaná do PC obsahují nejprve kód předmětu, dále data potřebná k vykreslení předmětu, rozměry, polohu a orientaci předmětu v prostoru. Celá aplikace vykreslí a vypíše všechny informace o detekovaném předmětu na obrazovku. Popis softwaru pro výpis a vykreslení dat (viz kapitola 6).

5. Nyní se provede přesun robota IRB 1400 do domovské pozice a vyšle synchronizační impuls, že se nachází v bezpečné vzdálenosti. Tento přesun se provádí z důvodu opuštění prostoru měření, kde bude provádět později pohyby robot obstarávající manipulační činnost.

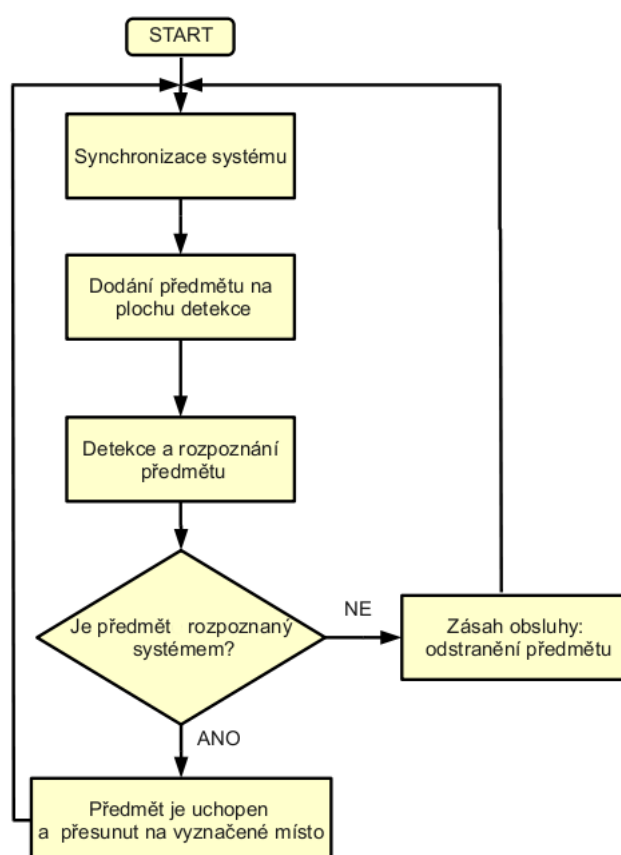
6. Dalším krokem je spojení řídicího systému IRB 1400 s IRB 140 přes rozhraní RS422, který mu odešle nejprve číselný kód předmětu, dále pozici v prostoru a výšku předmětu. Výška předmětu je důležitý údaj pro manipulačního robota z hlediska bezpečného odložení předmětu. Robot si vypočítá podle výšky předmětu z jaké vzdálenosti má předmět upustit na odkladní místo.



7. IRB 140 po přijetí dat vyše synchronizační impuls o svém počátku pohybu a začne se přibližovat po předem definované trajektorii k předmětu. Koncová trajektorie k uchopení detekovaného předmětu se vypočítá z příchozích dat, které určují přesnou pozici manipulačního předmětu.
8. Po uchopení předmětu přísavkou se robot IRB 140 začne přesouvat na předem definované místo určené kódem předmětu. Z přijaté výšky předmětu si robot dopočítá pozici koncového bodu trajektorie pro bezpečné odložení předmětu na odkládací místo.
9. Po upuštění předmětu se robot opět vrací do domovské pozice a informuje robota IRB 1400 pomocí synchronizačních impulsů, že může začít provádět znovu nový cyklus detekce předmětu.

Vytvořené programy pro oba roboty jsou součástí přiloženého CD k této práci, včetně kompletního videozáznamu průběhu činností popsaného robotického systému.

**Celý systém je znázorněn na vývojovém diagramu:**



Obrázek 54: Vývojový diagram robotického systému

## 7.2 Vyhodnocení spolehlivosti, přesnosti, použitelnosti a dosažitelnosti systému

V této části se budeme zabývat statistickým vyhodnocením funkčnosti celého systému. V předchozí kapitole jsme si vysvětlili aplikované principy rozpoznávání předmětu, které je potřeba nyní statisticky ověřit zda je navržená metoda úspěšná. Dosažené výsledky výpočtů z této části byly získány částečnou implementací statistických algoritmů do vizualizačního programu.

### Statisticky zkoumané hodnoty:

1. pravděpodobnost úspěšného rozpoznání tvaru předmětu
2. dosažená přesnost měření celého systému
3. výběrový rozptyl rozdílu skutečných a naměřených rozměrů předmětu
4. směrodatná odchylka rozdílu skutečných a naměřených rozměrů předmětu

Výsledky statistik jsou velice závislé na správnosti seřízení a kalibraci celého systému.

#### 7.2.1 Pravděpodobnost úspěšného rozpoznání předmětu

Jedná se o hodnotu udávající přehled, s jakou pravděpodobností systém rozpozná tvar předmětu a správně zařadí ho do definovaných kategorií. Pravděpodobnost správného rozpoznání předmětu se vypočítá jako:

$$\begin{aligned} p & - \text{pravděpodobnost jevu} \\ m & - \text{počet úspěšných pokusů rozpoznání předmětů} \\ n & - \text{počet všech pokusů o rozpoznání předmětů} \end{aligned}$$

$$p = \frac{m}{n} \quad (33)$$

Procentuální úspěšnost systému se vypočítá jako:

$$\text{úspěšnost}(\%) = p \cdot 100 \quad (34)$$

#### Přehled výsledků úspěšnosti rozpoznání tvaru předmětu:

Pro každý předmět bylo provedeno 50 měření, při kterých byl předmět umístěn s náhodnou polohou a orientací v oblasti detekční plochy. Umístění předmětu

prováděla obsluha systému. Ve výrobě by tuto pozici nahrazoval určitý druh automatického podavače.

#### **Kategorie 1:**

Válec: 100%

Koule: 100%

Kužel: 98%

#### **Kategorie 2:**

Krychle: 94%

Pravidelný čtyřboký jehlan: 92%

Hranol s podstavou čtverce: 94%

#### **Kategorie 3:**

Nepravidelný čtyřboký jehlan: 92%

Hranol s podstavou obdélníka: 96%

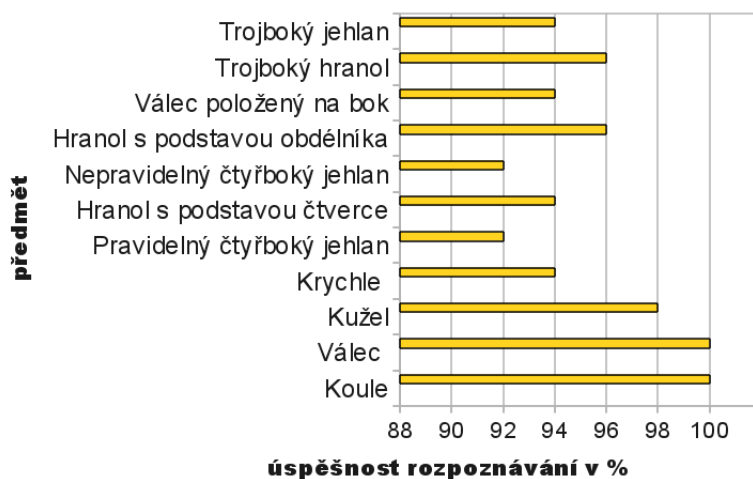
Válec položený na bok: 94%

#### **Kategorie 4:**

Trojboký hranol: 96%

Trojboký jehlan: 94%

**Graf úspěšnosti rozpoznání tvaru předmětů**



Obrázek 55: Zobrazení úspěšnosti rozpoznání pro jednotlivé předměty

**Celková úspěšnost rozpoznávání tvaru předmětu je: 95.45%.**

### 7.2.2 Dosažená přesnost měření celého systému

Dosaženou přesností měření systému budeme chápat jako průměrný rozdíl velikosti skutečných a naměřených rozměrů předmětu, který nazveme rozdíl rozměrů.

Přesné rozměry předmětů byly měřeny pomocí posuvného měřítka s přesností 0.1 mm. Opět pro každý předmět bylo provedeno 50 měření, ze kterých se vypočítaly rozdíly rozměrů naměřených posuvným měřítkem a robotickým systémem.

Dále budeme pro každý předmět určovat výběrový rozptyl a směrodatnou odchylku vypočítaných rozdílů rozměrů předmětů. Oba zkoumané parametry mají pro každý předmět opět 50 vypočítaných hodnot rozdílů měření.

**Výpočet výběrového rozptylu:** Rozptyl je definován jako střední hodnota kvadrátů odchylek od střední hodnoty. Určuje míru variability daného souboru dat.

$$\sigma^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2 \quad (35)$$

, kde  $x_i$  je hodnota  $i$ -tého pozorování,  $n$  je počet pozorování a  $\bar{x}$  je průměrná hodnota všech pozorování.

**Výpočet směrodatné odchylky:** Vyjadřuje rozptyl hodnot kolem střední hodnoty, tj. vypovídá o tom, jak se hodnoty od této střední hodnoty (průměru) liší, resp. jak hustě jsou kolem tohoto průměru seskupeny.

$$\sigma = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2} \quad (36)$$

, kde  $x_i$  je hodnota  $i$ -tého pozorování,  $n$  je počet pozorování a  $\bar{x}$  je průměrná hodnota všech pozorování.

#### Kategorie 1:

**Válec:** přesnost měření:  $\pm 0.7$  mm,

výběrový rozptyl: 0.04 mm,

směrodatná odchylka: 0.22 mm

**Koule:** přesnost měření:  $\pm 0.9$  mm

výběrový rozptyl: 0.08 mm,

směrodatná odchylka: 0.28 mm

**Kužel:** přesnost měření  $\pm 1.3$  mm

výběrový rozptyl: 0.15 mm,

směrodatná odchylka: 0.38 mm

## Kategorie 2:

**Krychle:** přesnost měření  $\pm 1.7$  mm

výběrový rozptyl: 0.15 mm,

směrodatná odchylka: 0.38 mm

**Pravidelný čtyřboký jehlan:** přesnost měření:  $\pm 1.9$  mm

výběrový rozptyl: 0.13 mm,

směrodatná odchylka: 0.37 mm

**Hranol s podstavou čtverce:** přesnost měření:  $\pm 1.6$  mm

výběrový rozptyl: 0.08 mm,

směrodatná odchylka: 0.29 mm

## Kategorie 3:

**Nepravidelný čtyřboký jehlan:** přesnost měření:  $\pm 1.9$  mm

výběrový rozptyl: 0.07 mm,

směrodatná odchylka: 0.27 mm

**Hranol s podstavou obdélníka:** přesnost měření:  $\pm 1.6$  mm

výběrový rozptyl: 0.1 mm,

směrodatná odchylka: 0.31 mm

**Válec položený na bok:** přesnost měření:  $\pm 1.8$  mm

výběrový rozptyl: 0.05 mm,

směrodatná odchylka: 0.24 mm

## Kategorie 4:

**Trojboký hranol:** přesnost měření:  $\pm 1.3$  mm

výběrový rozptyl: 0.08 mm,

směrodatná odchylka: 0.28 mm

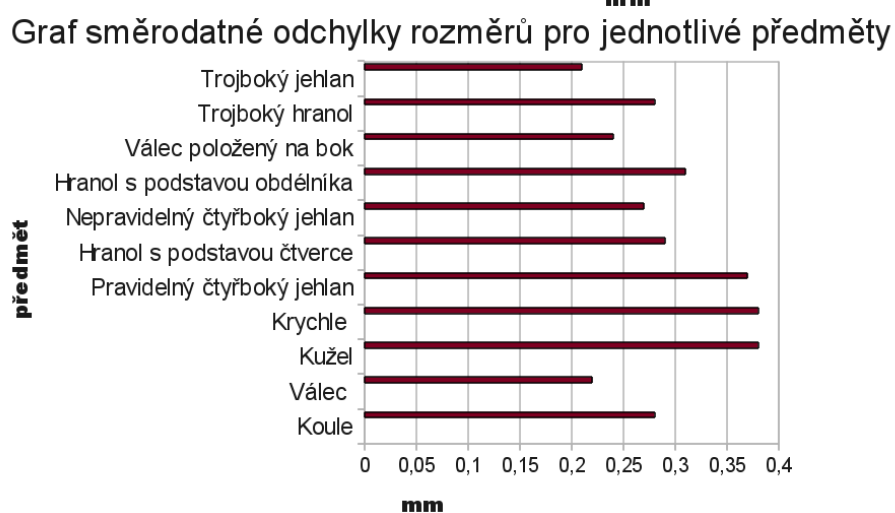
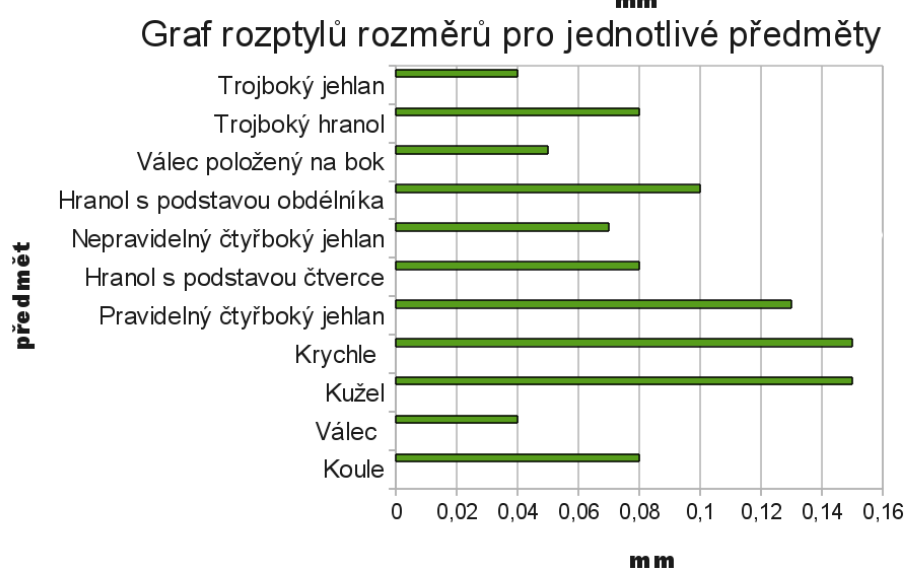
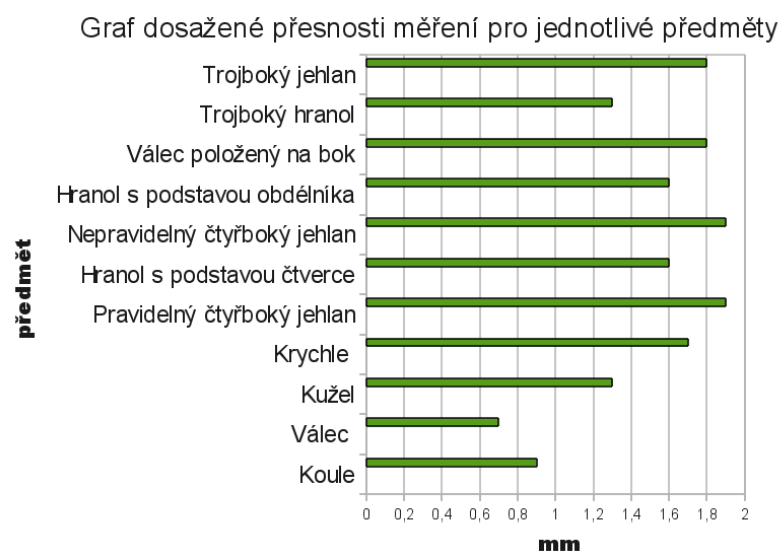
**Trojboký jehlan:** přesnost měření:  $\pm 1.8$  mm

výběrový rozptyl: 0.04 mm,

směrodatná odchylka: 0.21 mm

**Celková přesnost měření systémem je:  $\pm 1.5$  mm.**

Přesnost naměřených hodnot je závislá na správnosti seřízení celého systému. Do celkového výsledku měření se promítají zvolené způsoby měření, kdy jsme se snažili vyhnout nepřesnému měření pomocí centroidů a dále průměrováním všech spolu souvisejících stran. V neposlední řadě také zasahuje do výsledné přesnosti měření povaha konstrukce nástroje vytvořeného pro přichycení snímače k rameni robota.

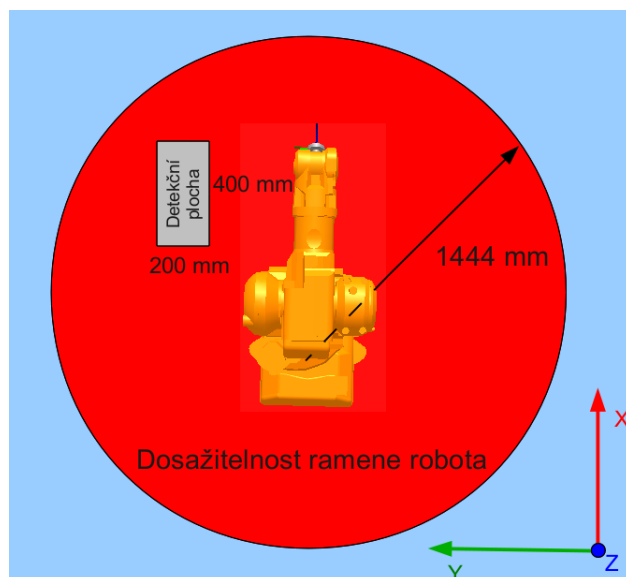


Obrázek 56: Přehled grafů zobrazující dosaženou přesnost, výběrový rozptyl a směrodatnou odchylku pro jednotlivé předměty

### 7.2.3 Dosažitelnost měřicího systému

Dosažitelnost systému je dána maximálním dostupným rozsahem ramene robota, ke kterému je připevněna konstrukce nesoucí snímač. Dosah ramene robota IRB 1400 provádějícího detekci je 1444 mm, více o technických parametrech nalezneme v kapitole (viz 3). Velikost detekční plochy je 200 x 400 mm, která je limitována rozpětím konstrukce nesoucí snímač. Rozpětí konstrukce je 230 mm, měřeno mezi oběma částmi snímače. Pro správnou funkčnost systému je důležité, aby byl celý detekční prostor uvnitř plochy vyznačující dosah ramene. Celá situace je přehledně nakreslena na obrázku (viz 57).

**Poznámka:** při detekci objektu se začne nejdříve prohledávat osa x, pro omezení velikosti delší strany (400 mm) detekční plochy.



Obrázek 57: Naznačení dosažitelnosti ramene robota a vyznačení plochy, na které byla prováděna detekce

## Závěr

Výsledkem diplomové práce je funkční robotický systém s částečnou inteligencí pro rozpoznávání základních tvarů předmětů v prostoru pomocí snímače MetraLight Rx series. Systém dále dokáže určit rozměry, orientaci a polohu detekovaného předmětu v prostoru. Dle zmíněných bodů zadání byla navržena a vytvořena koncová hlavička robota pro přichycení snímače MetraLight a pneumatických kleští. Rozšiřující částí celé práce je vytvoření vizualizačního programu v jazyku C++, kterým je možné pozorovat na PC naměřené hodnoty systému o rozpoznaném tvaru předmětu (poloha v prostoru, rozměry, orientace). Program dále také pro přehlednost vykreslí tvar detekovaného předmětu ve 2D (pohled shora v ose z) na obrazovku.

Systém rozpoznává následující tvary předmětů: válec, koule, kužel, krychle, pravidelný čtyřboký jehlan, hranol s podstavou čtverce, nepravidelný čtyřboký jehlan, hranol s podstavou obdélníku, válec položený na bok, trojboký hranol, trojboký jehlan.

Nerozpoznávané tvary předmětů: Z použitého zařízení jasně vyplývá, že tělesa konkrétního tvaru je nemožné rozpoznat. Dále z navržené metody by nebylo možné rozpoznat hranoly pěti a víceboké a dále všechna komolá tělesa. Řešení rozpoznání předchozích dvou tvarů předmětů bylo naznačeno na konci kapitoly věnované rozpoznávání tvarů předmětů. Systém nerozpoznává předměty různých tvarů průřezů v jejich spodní a horní části. To vychází z navržené metody rozpoznávání. Mezi předměty nerozpoznávané tímto systémem patří i ty, jejichž velikosti stran a celková výška je menší než 30 mm. To vychází z konstrukce snímače a navržené měřicí metody.

Navržený systém rozpoznávání předmětů dokáže s poměrně velkou úspěšností rozpoznávat základní tvary používaných předmětů v práci. Dosažená průměrná úspěšnost rozpoznávacího systému je 95.45%. Dosažená průměrná přesnost měření rozměrů systému je  $\pm 1.5$  mm. Čas potřebný pro detekci a rozpoznání jednoho předmětu je cca 40 vteřin.

Řešení rozpoznávání definovaných tvarů předmětů pomocí laserového snímače přináší hned několik výhod oproti použití průmyslovým systémům využívající kamery. Mezi hlavní výhody nesporně patří barevná i materiálová nezávislost detekovaných objektů a prostředí, ve kterém je prováděno rozpoznávání. Celý systém je díky laserovému snímači světelně nezávislý. Detekce předmětu může probíhat za zhoršených světelných podmínek nebo dokonce i za tmy. Zatím co s použitím průmyslových kamer, by za těchto podmínek nebylo možné realizovat



rozpoznávání. Další výhodou je i malá výpočetní náročnost tohoto systému, ve kterém neprobíhá zpracování složitých obrazových dat. Pro rozpoznání všech tvarů použitých předmětů by bylo zapotřebí využít více kamer.

Porovnání využití laserového snímače oproti speciálním průmyslovým systémům z hlediska finanční náročnosti. V průmyslu se nejčastěji používají tyto systémy:

- **PC systémy** - založené na klasických PC, do kterých jsou připojeny průmyslové kamery nebo digitální fotoaparáty. Při použití zdarma dostupného OpenSource softwaru pro zpracování obrazu vychází toto řešení levněji, než při použití snímače. Tento software je ovšem omezený svými funkcemi, a proto je použitelný jen na určité druhy úloh. Při využití placeného softwaru od renomovaných firem, které se zabývají zpracováním obrazu, je tato varianta finančně náročnější. Řešení pomocí PC systému je náročnější na údržbu a vlastní správu.
- **Compact Vision System (CVS)** - speciální průmyslové zařízení pro získání a zpracování obrazu. Je to autonomní zařízení bez klasického operačního systému s jednoúčelovým PC, ke kterému se připojí jedna či více průmyslových kamer. Toto řešení je spolehlivější a odolnější, než předchozí řešení. Finanční náročnost je řádově vyšší, než při použití laserového snímače.
- **Inteligentní kamery** - jedná se o podobné zařízení, jako je CVS, ovšem má možnost připojení pouze jedné kamery. Pro náročnější úlohy analýzy a rozpoznávání obrazu jsou však tyto prostředky značně omezené. Celková finanční náročnost je lehce vyšší, než při využití snímače pro tuto práci.

Z popsaných možností se jeví využití laserového snímače finančně realizovatelné. Levnějším řešením je pouze využití PC systému, který se ale potýká s problematikou real-time provozu operačního systému. Je nutné podotknout, že popsané řešení v diplomové práci se nabízí pro speciální aplikace, ve kterých se využijí přednosti navrženého systému za cenu větší časové náročnosti.

Vytvořený robotizovaný systém je možné využívat například pro třídění nebo výběr předmětů na základě určení jejich tvarů a rozměrů. Nasazení tohoto systému může být vhodné například ve slévárenském, sklářském nebo jiném výrobním provozu, kde je možné se setkat s tvary předmětů použitých v této práci. Celý robotický systém rozpoznávání předmětů byl úspěšně odzkoušen v laboratoři inteligentních robotů pod dohledem vedoucího práce.

Pro naprogramování robotického systému byl použit programovací jazyk Rapid pro roboty ABB ve vývojovém prostředí ProgramMaker. Při řešení práce bylo nutné funkčnost jednotlivých částí programu testovat na reálném systému. Pro vytvoření vizualizační aplikace byl dle potřeby nastudován programovací jazyk C++ a vývojové prostředí Borland C++ Builder. Pro pohyby robotů byl použit lineární pohyb vycházející z principu navržené metody rozpoznávání předmětu. Při řešení některých částí diplomové práce bylo použito pro zjednodušení situace simulačního softwaru RobotStudio 4.0, kterým TUL disponuje.

### **Při řešení práce bylo prostudováno a popsáno**

- Základy použitého robotického zařízení, především roboty ABB, včetně kompletního nastavení jejich datové komunikace v systému.
- Vlastnosti a možnosti využití laserového snímače MetraLight Rx series.
- Potřebné kinematické vztahy pro realizaci celé úlohy.
- Popis a vysvětlení vybraných instrukcí jazyka Rapid potřebných pro realizaci robotické aplikace.
- V praktické části se jednalo o klíčovou část celé práce zaměřenou na detekci a rozpoznávání předmětu.

Při řešení úlohy byl kladen důraz na dosažení co nejvyšší přesnosti měření, a to maximálním možným průměrováním rozměrů spolu souvisejících. Avšak celkový pohled na dosažené výsledky není úplně ideální, vzhledem k použitému zařízení. Nyní se pokusíme nastínit pár možností, jak vylepšit funkčnost stávajícího návrhu systému.

Z hlediska zlepšení přesnosti měření celého systému by bylo vhodné zhotovit lepší hlavici pro přichycení snímače MetraLight. Ta stávající je postačující, avšak má řadu výhod i nevýhod. K výhodám neodmyslitelně patří cena celé konstrukce, dále možnost libovolného nastavení a natočení díky rozkládacím dílům. Mezi hlavní nevýhody patří nedostatečná tuhost materiálu konstrukce, která přináší určitou pružnost. To je dáno mimo jiné i tím že je složena z několika částí. Při návrhu kvalitnější koncové hlavice by bylo vhodné se nejvíce věnovat přesnosti rozměrů vytvořeného nástroje a vhodné volbě materiálu, ze kterého bude vyroben.

Pro snížení časové náročnosti detekce předmětu se nabízejí tři možnosti.

**První možnost** je využití rychlejšího řídicího systému robota, který by dovoľoval kontrolovat snímač vícekrát, než 4-krát za sekundu.

**Druhou možnost** je zmenšení množiny rozpoznávaných předmětů, pro kterou by stačilo vygenerovat méně detekčních přímek a tím snížit počet trajektorií.

**Třetí možnost** je využít více snímačů pro detekci předmětu nebo použít snímač s větší detekční plochou.

Způsobů, jak navázat a rozšířit tuto práci, se nabízí hned několik. Jednou z hlavních možností je rozšířit již navržené metody detekce a rozpoznávání předmětu. Rozšíření by především obsahovalo vytvoření algoritmů pro rozpoznávání dalších tvarů předmětů, které by přineslo zvětšení množiny, kterou systém rozpoznává. Dále má celý systém stále rezervy v přesnosti měření, a proto by bylo zapotřebí vhodněji navrhnout koncovou hlavici pro přichycení snímače MetraLight. Celou aplikaci by dále ještě bylo možné vylepšit využitím více snímačů připevněných na koncové hlavici nebo samostatně na své nosné konstrukci. Jako rozšiřující snímače mohou být využity buď opět laserové, jako v této práci, nebo kombinace jiných například ultrazvukových popřípadě optických. Toto rozšíření by přineslo vyšší možnosti použitelnosti rozpoznávacího systému. Dále by se zlepšila dosažená přesnost při měření a také snížení časové náročnosti celého cyklu detekce a rozpoznávání. V neposlední řadě by také bylo vhodné tento systém upravit na plně autonomní, především po stránce řešení stavu systému, kdy předmět není rozpoznán. Stav, kdy předmět není rozpoznán prozatím řeší obsluha systému a neznámý předmět manuálně odstraní z plochy určené k detekci.

## Literatura

- [1] METRALIGHT REV.001 USER'S GUIDE *Firemní dokumentace ke snímači MetraLight*. <http://www.metrallight.com>
- [2] ROBOTSTUDIO 3.1 USER'S GUIDE. *Firemní dokumentace k programu RobotStudio, Švédsko 2004*.
- [3] RAPID REFERENCE MANUAL. *Firemní dokumentace k programu Rapid, Švédsko 2004*.
- [4] PRODUCT MANUAL IRB140. *Firemní dokumentace k robotu IRB140, Švédsko 2004*.
- [5] PŠENIČKA, RADEK A TVRZNÍK, MICHAL. *Demonstrační aplikace v RobotStudiu*. Ročníkový projekt, TUL FM 2007, 40s. ČR.
- [6] PŠENIČKA, RADEK A TVRZNÍK, MICHAL. *Identifikace předmětů pomocí snímače MetraLight pro průmyslové roboty*. Semestrální projekt, TUL FM 2009, 30s. ČR.
- [7] PŠENIČKA, RADEK. *Vzájemná koordinace pohybu dvou robotů*. Bakalářská práce, TUL FM 2008, 60s. ČR.
- [8] VÁVRA, Václav a LOSOS, Zdeněk. *Odvození transformačních matic pro různé typy rotace* [on-line]. Aktualizováno: 11.1.2007. [cit. 5. 5. 2010]. Dostupné na: [http://mineralogie.sci.muni.cz/kap\\_1\\_3\\_symetrie/rotace\\_priklad.htm](http://mineralogie.sci.muni.cz/kap_1_3_symetrie/rotace_priklad.htm).
- [9] KADLEC VÁCLAV. *Učíme se programovat v Borland C++ Builder a jazyce C++*. 1. vydání. Praha, Computer Press, 2002. 80-7226-550-4
- [10] HOLAN TOMÁŠ, NERUDA ROMAN. *C++ Builder v příkladech*. 1. vydání. Praha, BEN, 2002. 80-7300-042-3
- [11] MATELA, L. Počítačové zpracování obrazu. (přednáška) Liberec : Fakulta mechatroniky, informatiky a mezioborových studií TUL, 13.4.2009
- [12] NOUZA, J. Metody rozhodování a klasifikace. (přednáška) Liberec : Fakulta mechatroniky, informatiky a mezioborových studií TUL, 19.3.2008

# Přílohy

## 1. CD

Obsah CD:

- Literatura
  - User's Guide 4.0.80.pdf
  - Rapid Reference 4.0.80.pdf
  - Product Manual IRB 140 3HAC 7564-1 rev.1 M2000.pdf
  - Product Manual IRB 1400 3HAC 7617-1 rev. 1 M2000.pdf
  - MetraLight REV.001 User's Guide.pdf
- Zdrojové kódy robotů
  - IRB140.prg
  - IRB1400.prg
- Program pro vizualizaci dat
  - zdrojové kódy
  - spustitelný soubor vizualizační aplikace
- Video výsledné aplikace
  - video.wmv